OMOP Common Data Model Specifications

Christian Reich, Patrick Ryan, Rimma Belenkaya, Karthik Natarajan and Clair Blacketer

2018-10-10

Contents

1 License

2 Background

The Role of the Common Data Model
Design Principles
Data Model Conventions

The Observational Medical Outcomes Partnership (OMOP) was a public-private partnership established to inform the appropriate use of observational healthcare databases for studying the effects of medical products. Over the course of the 5-year project and through its community of researchers from industry, government, and academia, OMOP successfully achieved its aims to:

- Conduct methodological research to empirically evaluate the performance of various analytical methods on their ability to identify true associations and avoid false findings
- Develop tools and capabilities for transforming, characterizing, and analysing disparate data sources across the health care delivery spectrum
- Establish a shared resource so that the broader research community can collaboratively advance the science

The results of OMOP's research has been widely published and presented at scientific conferences, including annual symposia.

The OMOP Legacy continues…

The community is actively using the OMOP Common Data Model for their various research purposes. Those tools will continue to be maintained and supported, and information about this work is available in the public domain.

The Observational Health Data Sciences and Informatics (OHDSI) has been established as a multi-stakeholder, interdisciplinary collaborative to create open-source solutions that bring out the value of observational health data through large-scale analytics. The OHDSI collaborative includes all of the original OMOP research investigators, and will develop its tools using the OMOP Common Data Model. Learn more at ohdsi.org.

The OMOP Common Data Model will continue to be an open-source community standard for observational healthcare data. The model specifications and associated work products will be placed in the public domain, and the entire research community is encouraged to use these tools to support everybody's own research activities.

2.1 The Role of the Common Data Model

No single observational data source provides a comprehensive view of the clinical data a patient accumulates while receiving healthcare, and therefore none can be sufficient to meet all expected outcome analysis needs. This explains the need for assessing and analyzing multiple data sources concurrently using a common data standard. This standard is provided by the OMOP Common Data Model (CDM).

The CDM is designed to support the conduct of research to identify and evaluate associations between interventions (drug exposure, procedures, healthcare policy changes etc.) and outcomes caused by these interventions (condition occurrences, procedures, drug exposure etc.). Outcomes can be efficacious (benefit) or adverse (safety risk). Often times, specific patient cohorts (e.g., those taking a certain drug or suffering from a certain disease) may be defined for treatments or outcomes, using clinical events (diagnoses, observations, procedures, etc.) that occur in predefined temporal relationships to each other. The CDM, combined with its standardized content (via the Standardized Vocabularies), will ensure that research methods can be systematically applied to produce meaningfully comparable and reproducible results.

2.2 Design Principles

The CDM is designed to include all observational health data elements (experiences of the patient receiving health care) that are relevant for analysis use cases to support the generation of reliable scientific evidence about disease natural history, healthcare delivery, effects of medical interventions, the identification of demographic information, health care interventions and outcomes.

Therefore, the CDM is designed to store observational data to allow for research, under the following principles:

- **Suitability for purpose:** The CDM aims to provide data organized in a way optimal for analysis, rather than for the purpose of addressing the operational needs of health care providers or payers.
- **Data protection:** All data that might jeopardize the identity and protection of patients, such as names, precise birthdays etc. are limited. Exceptions are possible where the research expressly requires more detailed information, such as precise birth dates for the study of infants.
- **Design of domains:** The domains are modeled in a person-centric relational data model, where for each record the identity of the person and a date is captured as a minimum.
- **Rationale for domains:** Domains are identified and separately defined in an entity-relationship model if they have an analysis use case and the domain has specific attributes that are not otherwise applicable. All other data can be preserved as an observation in an entity-attribute-value structure.

- **Standardized Vocabularies:** To standardize the content of those records, the CDM relies on the Standardized Vocabularies containing all necessary and appropriate corresponding standard healthcare concepts.
- **Reuse of existing vocabularies:** If possible, these concepts are leveraged from national or industry standardization or vocabulary definition organizations or initiatives, such as the National Library of Medicine, the Department of Veterans' Affairs, the Center of Disease Control and Prevention, etc.
- **Maintaining source codes:** Even though all codes are mapped to the Standardized Vocabularies, the model also stores the original source code to ensure no information is lost.
- **Technology neutrality:** The CDM does not require a specific technology. It can be realized in any relational database, such as Oracle, SQL Server etc., or as SAS analytical datasets.
- **Scalability:** The CDM is optimized for data processing and computational analysis to accommodate data sources that vary in size, including databases with up to hundreds of millions of persons and billions of clinical observations.
- **Backwards compatibility:** All changes from previous CDMs are clearly delineated in the github repository. Older versions of the CDM can be easily created from the CDMv5, and no information is lost that was present previously.

2.3 Data Model Conventions

There are a number of implicit and explicit conventions that have been adopted in the CDM. Developers of methods that run against the CDM need to understand these conventions.

2.3.1 General conventions of schemas

New to CDM v6.0 is the concept of schemas. This allows for more separation between read-only and writeable tables. The clinical data, event, and vocabulary tables are in the 'CDM' schema and tables that need to be manipulated by web-based tools or end users have moved to the 'Results' schema. Currently the only two tables in the 'Results' schema are COHORT and COHORT_DEFINITON, though likely more will be added over the course of v6.0 point releases.

2.3.2 General conventions of data tables

The CDM is platform-independent. Data types are defined generically using ANSI SQL data types (VARCHAR, INTEGER, FLOAT, DATE, DATETIME, CLOB). Precision is provided only for VARCHAR. It reflects the minimal required string length and can be expanded within a CDM instantiation. The CDM does not prescribe the date and datetime format. Standard queries against CDM may vary for local instantiations and date/datetime configurations.

In most cases, the first field in each table ends in '_ID', containing a record identifier that can be used as a foreign key in another table.

2.3.3 General conventions of fields

Variable names across all tables follow one convention:

Notation	Description
_SOURCE_VALUE	Verbatim information from the source data, typically used in ETL to map to CONCEPT_ID, and not to be used by any standard analytics. For example, CONDITION_SOURCE_VALUE = '787.02' was the ICD-9 code captured as a diagnosis from the administrative claim.

Notation	Description
_ID	Unique identifiers for key entities, which can serve as foreign keys to establish relationships across entities. For example, PERSON_ID uniquely identifies each individual. VISIT_OCCURRENCE_ID uniquely identifies a PERSON encounter at a point of care.
_CONCEPT_ID	Foreign key into the Standardized Vocabularies (i.e. the standard_concept attribute for the corresponding term is true), which serves as the primary basis for all standardized analytics. For example, CONDITION_CONCEPT_ID = 31967 contains the reference value for the SNOMED concept of 'Nausea'
_SOURCE_CONCEPT_ID	Foreign key into the Standardized Vocabularies representing the concept and terminology used in the source data, when applicable. For example, CONDITION_SOURCE_CONCEPT_ID = 45431665 denotes the concept of 'Nausea' in the Read terminology; the analogous CONDITION_CONCEPT_ID might be 31967, since SNOMED-CT is the Standardized Vocabulary for most clinical diagnoses and findings.
_TYPE_CONCEPT_ID	Delineates the origin of the source information, standardized within the Standardized Vocabularies. For example, DRUG_TYPE_CONCEPT_ID can allow analysts to discriminate between 'Pharmacy dispensing' and 'Prescription written'

2.3.4 Representation of content through Concepts

In CDM data tables the meaning of the content of each record is represented using Concepts. Concepts are stored with their CONCEPT_ID as foreign keys to the CONCEPT table in the Standardized Vocabularies, which contains Concepts necessary to describe the healthcare experience of a patient. If a Standard Concept does not exist or cannot be identified, the Concept with the CONCEPT_ID 0 is used, representing a non-existing or un-mappable concept.

Records in the CONCEPT table contain all the detailed information about it (name, domain, class etc.). Concepts, Concept Relationships and other information relating to Concepts is contained in the tables of the Standardized Vocabularies.

2.3.5 Difference between Concept IDs and Source Values

Many tables contain equivalent information multiple times: As a Source Value, a Source Concept and as a Standard Concept.

- Source Values contain the codes from public code systems such as ICD-9-CM, NDC, CPT-4 etc. or locally controlled vocabularies (such as F for female and M for male) copied from the source data. Source Values are stored in the *_SOURCE_VALUE fields in the data tables.
- Concepts are CDM-specific entities that represent the meaning of a clinical fact. Most concepts are based on code systems used in healthcare (called Source Concepts), while others were created de-novo (CONCEPT_CODE = 'OMOP generated'). Concepts have unique IDs across all domains.
- Source Concepts are the concepts that represent the code used in the source. Source Concepts are only used for common healthcare code systems, not for OMOP-generated Concepts. Source Concepts are stored in the *_SOURCE_CONCEPT_ID field in the data tables.
- Standard Concepts are those concepts that are used to define the unique meaning of a clinical entity. For each entity there is one Standard Concept. Standard Concepts are typically drawn from existing public vocabulary sources. Concepts that have the equivalent meaning to a Standard Concept are mapped to the Standard Concept. Standard Concepts are referred to in the _CONCEPT_ID field of the data tables.

Source Values are only provided for convenience and quality assurance (QA) purposes. Source Values and Source Concepts are optional, while Standard Concepts are mandatory. Source Values may contain information that is only meaningful in the context of a specific data source.

2.3.6 Difference between general Concepts and Type Concepts

Type Concepts (ending in _TYPE_CONCEPT_ID) and general Concepts (ending in _CONCEPT_ID) are part of many tables. The former are special Concepts with the purpose of indicating where the data are derived from in the source. For example, the Type Concept field can be used to distinguish a DRUG_EXPOSURE record that is derived from a pharmacy-dispensing claim from one indicative of a prescription written in an electronic health record (EHR).

2.3.7 Time span of available data

Data tables for clinical data contain a datetime stamp (ending in _DATETIME, _START_DATETIME or _END_DATETIME), indicating when that clinical event occurred. As a rule, no record can be outside of a valid OBSERVATION_PERIOD time period. Clinical information that relates to events that happened prior to the first OBSERVATION_PERIOD will be captured as a record in the OBSERVATION table as 'Medical history' (CONCEPT_ID = 43054928), with the OBSERVATION_DATETIME set to the first OBSERVATION_PERIOD_START_DATE of that patient, and the VALUE_AS_CONCEPT_ID set to the corresponding CONCEPT_ID for the condition/drug/procedure that occurred in the past. No data occurring after the last OBSERVATION_PERIOD_END_DATE can be valid records in the CDM. * When mapping source data to the CDM, if time is unknown the default time of 00:00:00 should be chosen. If a time of 00:00:00 is given in the source data it should be shifted to 00:00:01 (THEMIS issue #10).

2.3.8 Content of each table

For the tables of the main domains of the CDM it is imperative that concepts used are strictly limited to the domain. For example, the CONDITION_OCCURRENCE table contains only information about conditions (diagnoses, signs, symptoms), but no information about procedures. Not all source coding schemes adhere to such rules. For example, ICD-9-CM codes, which contain mostly diagnoses of human disease, also contain information about the status of patients having received a procedure. The ICD-9-CM code V20.3 'Newborn health supervision' defines a continuous procedure and is therefore stored in the PROCEDURE_OCCURRENCE table.

2.3.9 Differentiating between Source Values, Source Concept Ids, and Standard Concept Ids

Each table contains fields for Source Values, Source Concept Ids, and Standard Concept Ids.

- Source Values are fields to maintain the verbatim information from the source database, stored as unstructured text, and are generally not to be used by any standardized analytics. There is no standardization for these fields and these columns can be used as the local CDM builders see fit. A typical example would be an ICD-9 code without the decimal from an administrative claim as condition_source_value = '78702' which is how it appeared in the source (THEMIS issue #15).
- Source Concept Ids provide a repeatable representation of the source concept, when the source data are drawn from a commonly-used, internationally-recognized vocabulary that has been distributed with the OMOP Common Data Model. Specific use cases where source vocabulary-specific analytics are required can be accommodated by the use of the *_SOURCE_CONCEPT_ID fields, but these are generally not applicable across disparate data sources. The standard *_CONCEPT_ID fields are **strongly suggested** to be used in all standardized analytics, as specific vocabularies have been established within each data domain to facilitate standardization of both structure and content within the OMOP Common Data Model.

The following provide conventions for processing source data using these three fields in each domain:

When processing data where the source value is either free text or a reference to a coding scheme that is not contained within the Standardized Vocabularies:

- Map all Source Values to the corresponding Standard CONCEPT_IDs. Store the CONCEPT_IDs in the TARGET_CONCEPT_ID field of the SOURCE_TO_CONCEPT_MAP table.
  - If a CONCEPT_ID is not available for the source code, the TARGET_CONCEPT_ID field is set to 0.

When processing your data where Source Value is a reference to a coding scheme contained within the Standardized Vocabularies:

- Find all CONCEPT_IDs in the Source Vocabulary that correspond to your Source Values. Store the result in the SOURCE_CONCEPT_ID field.
  - If the source code follows the same formatting as the distributed vocabulary, the mapping can be directly obtained from the CONCEPT table using the CONCEPT_CODE field.
  - If the source code uses alternative formatting (ex. format has removed decimal point from ICD-9 codes), you will need to perform the formatting transformation within the ETL. In this case, you may wish to store the mappings from original codes to SOURCE_CONCEPT_IDs in the SOURCE_TO_CONCEPT_MAP table.
  - If the source code is not found in a vocabulary, the SOURCE_CONCEPT_ID field is set to 0
- Use the CONCEPT_RELATIONSHIP table to identify the Standard CONCEPT_ID that corresponds to the SOURCE_CONCEPT_ID in the domain.
  - Each SOURCE_CONCEPT_ID can have 1 or more Standard CONCEPT_IDs mapped to it. Each Standard CONCEPT_ID belongs to only one primary domain but when a source CONCEPT_ID maps to multiple Standard CONCEPT_IDs, it is possible for that SOURCE_CONCEPT_ID to result in records being produced across multiple domains. For example, ICD-10-CM code Z34.00 'Encounter for supervision of normal first pregnancy, unspecified trimester' will be mapped to the SNOMED concept 'Routine antenatal care' in the procedure domain and the concept in the condition domain 'Primagravida'. It is also possible for one SOURCE_CONCEPT_ID to map to multiple Standard CONCEPT_IDs within the same domain. For example, ICD-9-CM code 070.43 'Hepatitis E with hepatic coma' maps to the SNOMED concept for 'Acute hepatitis E' and a second SNOMED concept for 'Hepatic coma', in which case multiple CONDITION_OCCURRENCE records will be generated for the one source value record.
  - If the SOURCE_CONCEPT_ID is not mappable to any Standard CONCEPT_ID, the _CONCEPT_ID field is set to 0.
- Write the data record into the table(s) corresponding to the domain of the Standard CONCEPT_ID(s).
  - If the Source Value has a SOURCE_CONCEPT_ID but the SOURCE_CONCEPT_ID is not mapped to a Standard CONCEPT_ID, then the domain for the data record, and hence it's table location, is determined by the DOMAIN_ID field of the CONCEPT record the SOURCE_CONCEPT_ID refers to. The Standard _CONCEPT_ID is set to 0.
  - If the Source Value cannot be mapped to a SOURCE_CONCEPT_ID or Standard CONCEPT_ID, then direct the data record to the most appropriate CDM domain based on your local knowledge of the intent of the source data and associated value. For example, if the un-mappable Source Value came from a 'diagnosis' table then, in the absence of other information, you may choose to record that fact in the CONDITION_OCCURRENCE table.

Each Standard CONCEPT_ID field has a set of allowable CONCEPT_ID values. The allowable values are defined by the domain of the concepts. For example, there is a domain concept of 'Gender', for which there are only two allowable standard concepts of practical use (8507 - 'Male', 8532- 'Female') and one allowable generic concept to represent a standard notion of 'no information' (concept_id = 0). This 'no information' concept should be used when there is no mapping to a standard concept available or if there is no information available for that field. The exceptions are MEASUREMENT.VALUE_AS_CONCEPT_ID, OBSERVATION.VALUE_AS_CONCEPT_ID, MEASUREMENT.UNIT_CONCEPT_ID, OBSER-

VATION.UNIT_CONCEPT_ID, MEASUREMENT.OPERATOR_CONCEPT_ID, and OBSERVATION.MODIFIER_CONCEPT_ID, which can be NULL if the data do not contain the information (THEMIS issue #11).

There is no constraint on allowed CONCEPT_IDs within the SOURCE_CONCEPT_ID fields.

2.3.10 Custom SOURCE_TO_CONCEPT_MAPs

When the source data uses coding systems that are not currently in the Standardized Vocabularies (e.g. ICPC codes for diagnoses), the convention is to store the mapping of such source codes to Standard Concepts in the SOURCE_TO_CONCEPT_MAP table. The codes used in the data source can be recorded in the SOURCE_VALUE fields, but no SOURCE_CONCEPT_ID will be available.

Custom source codes are not allowed to map to Standard Concepts that are marked as invalid.

3 Glossary of Terms

Glossary of Terms

Term	Abbr.	Description
Ancestor		The higher level Concept in a hierarchical relationship. Note that ancestors and descendants can be many levels apart from each other.
Average Wholesale Price	AWP	The price manufacturers set for prescription drugs to be purchased at the wholesale level to pharmacies and healthcare provider.
Centers for Disease Control and Prevention	CDC	The Centers for Disease Control and Prevention is a United States federal agency under the Department of Health and Human Services. It works to protect public health and safety by providing information to enhance health decisions.
Common Data Model	CDM	The CDM intends to facilitate observational analyses of disparate healthcare databases. The CDM defines table structures for each of the data entities (e.g., Persons, Visit Occurrence, Drug Exposure, Condition Occurrence, Observation, Procedure Occurrence, etc.). It includes observational data elements that are relevant to identifying exposure to various treatments and defining condition occurrence. The CDM includes both the Standardized Vocabularies of terms and the entity domain tables.
Concept		A concept is the basic unit of information. Concepts may be grouped into a given domain. A concept is a unique term that has a unique and static identifier/name, belongs to a domain, and may exist in relation to other concepts. The vertical relationships consist of “is a” statements that form a logical hierarchy. In general, concepts above a given concept are referred to as ancestors and those below as descendants.

Term	Abbr.	Description
Conceptual Data Model		A conceptual data model is a map of concepts and their relationships. This describes the semantics of an organization and represents a series of assertions about its nature. Specifically, it describes the things of significance to an organization (entity classes), about which it is inclined to collect information, and characteristics of (attributes) and associations between pairs of those things of significance (relationships).
Data mapping		It is the data element mappings between two distinct data models, terminologies, or concepts. Data mapping is the process of creating data element mappings between two distinct data models. Data mapping is used as a first step for a wide variety of data integration tasks.
Demographics		Demographics refer to selected characteristics of persons. Demographics may include data such as race, age, sex, date of birth, location, etc.
Descendant		The lower level Concept in a hierarchical relationship. Note that ancestors and descendants can be many levels apart from each other.
Design Principle		An organized arrangement of one or more elements or principles for a purpose. It identifies core principles and best practices to assist developers to produce software. Thoroughly understanding the goals of stakeholders and designing systems with those goals in mind are the best approaches to successfully deliver results.
Electronic Health Record	EHR	Electronic health record refers to an individual person's medical record in digital format. It may be made up of electronic medical records from many locations and/or sources. The EHR is a longitudinal electronic record of person health information generated by one or more encounters in any care delivery setting. Included in this information are person demographics, progress notes, problems, medications, vital signs, past medical history, immunizations, laboratory data and radiology reports.
Electronic Medical Record	EMR	An electronic medical record is a computerized medical record created in an organization that delivers care, such as a hospital or outpatient setting. Electronic medical records tend to be a part of a local stand-alone health information system that allows storage, retrieval and manipulation of records. This document will reference EHR moving forward even if specific data source might internally use EMR definition.
Extract Transform Load	ETL	Process of getting data out of one data store (Extract), modifying it (Transform), and inserting it into a different data store (Load).

Term	Abbr.	Description
Health Insurance Portability and Accountability Act	HIPAA	A federal law that was designed to allow portability of health insurance between jobs. In addition, it required the creation of a federal law to protect personally identifiable health information; if that did not occur by a specific date (which it did not), HIPAA directed the Department of Health and Human Services (DHHS) to issue federal regulations with the same purpose. DHHS has issued HIPAA privacy regulations (the HIPAA Privacy Rule) as well as other regulations under HIPAA.
Logical Data Model		Logical data models are graphical representation of the business requirements. They describe the things of importance to an organization and how they relate to one another, as well as business definitions and examples. The logical data model can be validated and approved by a business representative, and can be the basis of physical database design.
Primary Care Provider	PCP	A health care provider designated as responsible to provide general medical care to a patient, including evaluation and treatment as well as referral to specialists.
Protected Health Information	PHI	Protected health information under HIPAA includes any individually identifiable health information. Identifiable refers not only to data that is explicitly linked to a particular individual (that's identified information). It also includes health information with data items which reasonably could be expected to allow individual identification. De-identified information is that from which all potentially identifying information has been removed.
Terminology		Technical or special terms used in a business or special subject area.
Vocabulary		A computerized list (as of items of data or words) used for reference (as for information retrieval or word processing).

4 Standardized Vocabularies

CONCEPT
VOCABULARY
DOMAIN
CONCEPT_CLASS
CONCEPT_RELATIONSHIP
RELATIONSHIP
CONCEPT_SYNONYM
CONCEPT_ANCESTOR
SOURCE_TO_CONCEPT_MAP
DRUG_STRENGTH

These tables contain detailed information about the Concepts used in all of the CDM fact tables. The content of the Standardized Vocabularies tables is not generated anew by each CDM implementation. Instead, it is maintained centrally as a service to the community.

A number of assumptions were made for the design of the Standardized Vocabularies tables:

- There is one design which will accommodate all different source terminologies and classifications.
- All terminologies are loaded into the CONCEPT table.
- The key is a newly created concept_id, not the original code of the terminology, because source codes are not unique identifiers across terminologies.
- Some Concepts are declared Standard Concepts, i.e. they are used to represent a certain clinical entity in the data. All Concepts may be Source Concepts; they represent how the entity was coded in the source. Standard Concepts are identified through the standard_concept field in the CONCEPT table.
- Records in the CONCEPT_RELATIONSHIP table define semantic relationships between Concepts. Such relationships can be hierarchical or lateral.
- Records in the CONCEPT_RELATIONSHIP table are used to map Source codes to Standard Concepts, replacing the mechanism of the SOURCE_TO_CONCEPT_MAP table used in prior Standardized Vocabularies versions. The SOURCE_TO_CONCEPT_MAP table is retained as an optional aid to bookkeeping codes not found in the Standardized Vocabularies.
- Chains of hierarchical relationships are recorded in the CONCEPT_ANCESTOR table. Ancestry relationships are only recorded between Standard Concepts that are valid (not deprecated) and are connected through valid and hierarchical relationships in the RELATIONSHIP table (flag DEFINES_ANCESTRY).

The advantage of this approach lies in the preservation of codes and relationships between them without adherence to the multiple different source data structures, a simple design for standardized access, and the optimization of performance for analysis. Navigation among Standard Concepts does not require knowledge of the source vocabulary. Finally, the approach is scalable and future vocabularies can be integrated easily. On the other hand, extensive transformation of source data to the Vocabulary is required and not every source data structure and original source hierarchy can be retained.

Below is an entity-relationship diagram highlighting the tables within the Vocabulary portion of the OMOP Common Data Model:

4.1 CONCEPT

The Standardized Vocabularies contains records, or Concepts, that uniquely identify each fundamental unit of meaning used to express clinical information in all domain tables of the CDM. Concepts are derived from vocabularies, which represent clinical information across a domain (e.g. conditions, drugs, procedures) through the use of codes and associated descriptions. Some Concepts are designated Standard Concepts, meaning these Concepts can be used as normative expressions of a clinical entity within the OMOP Common Data Model and within standardized analytics. Each Standard Concept belongs to one domain, which defines the location where the Concept would be expected to occur within data tables of the CDM.

Concepts can represent broad categories (like 'Cardiovascular disease'), detailed clinical elements ('Myocardial infarction of the anterolateral wall') or modifying characteristics and attributes that define Concepts at various levels of detail (severity of a disease, associated morphology, etc.).

Records in the Standardized Vocabularies tables are derived from national or international vocabularies such as SNOMED-CT, RxNorm, and LOINC, or custom Concepts defined to cover various aspects of observational data analysis. For a detailed description of these vocabularies, their use in the OMOP CDM and their relationships to each other please refer to the specifications.

Field	Required	Type	Description
concept_id	Yes	integer	A unique identifier for each Concept across all domains.
concept_name	Yes	varchar(255)	An unambiguous, meaningful and descriptive name for the Concept.

Field	Required	Type	Description
domain_id	Yes	varchar(20)	A foreign key to the DOMAIN table the Concept belongs to.
vocabulary_id	Yes	varchar(20)	A foreign key to the VOCABULARY table indicating from which source the Concept has been adapted.
concept_class_id	Yes	varchar(20)	The attribute or concept class of the Concept. Examples are 'Clinical Drug', 'Ingredient', 'Clinical Finding' etc.
standard_concept	No	varchar(1)	This flag determines where a Concept is a Standard Concept, i.e. is used in the data, a Classification Concept, or a non-standard Source Concept. The allowables values are 'S' (Standard Concept) and 'C' (Classification Concept), otherwise the content is NULL.
concept_code	Yes	varchar(50)	The concept code represents the identifier of the Concept in the source vocabulary, such as SNOMED-CT concept IDs, RxNorm RXCUIs etc. Note that concept codes are not unique across vocabularies.
valid_start_date	Yes	date	The date when the Concept was first recorded. The default value is 1-Jan-1970, meaning, the Concept has no (known) date of inception.
valid_end_date	Yes	date	The date when the Concept became invalid because it was deleted or superseded (updated) by a new concept. The default value is 31-Dec-2099, meaning, the Concept is valid until it becomes deprecated.
invalid_reason	No	varchar(1)	Reason the Concept was invalidated. Possible values are D (deleted), U (replaced with an update) or NULL when valid_end_date has the default value.

4.1.1 Conventions

Concepts in the Common Data Model are derived from a number of public or proprietary terminologies such as SNOMED-CT and RxNorm, or custom generated to standardize aspects of observational data. Both types of Concepts are integrated based on the following rules:

No.	Convention Description
1	All Concepts are maintained centrally by the CDM and Vocabularies Working Group. Additional concepts can be added, as needed, upon request.

No.	Convention Description
2	For all Concepts, whether they are custom generated or adopted from published terminologies, a unique numeric identifier concept_id is assigned and used as the key to link all observational data to the corresponding Concept reference data.
3	The concept_id of a Concept is persistent, i.e. stays the same for the same Concept between releases of the Standardized Vocabularies.
4	A descriptive name for each Concept is stored as the Concept Name as part of the CONCEPT table. Additional names and descriptions for the Concept are stored as Synonyms in the CONCEPT_SYNONYM table.
5	Each Concept is assigned to a Domain. For Standard Concepts, these is always a single Domain. Source Concepts can be composite or coordinated entities, and therefore can belong to more than one Domain. The domain_id field of the record contains the abbreviation of the Domain, or Domain combination. Please refer to the Standardized Vocabularies specification for details of the Domain Assignment.
6	For details of the Vocabularies adopted for use in the OMOP CDM refer to the Standardized Vocabularies specification.
7	Concept Class designation are attributes of Concepts. Each Vocabulary has its own set of permissible Concept Classes, although the same Concept Class can be used by more than one Vocabulary. Depending on the Vocabulary, the Concept Class may categorize Concepts vertically (parallel) or horizontally (hierarchically). See the specification of each vocabulary for details.
8	Concept Class attributes should not be confused with Classification Concepts. These are separate Concepts that have a hierarchical relationship to Standard Concepts or each other, while Concept Classes are unique Vocabulary-specific attributes for each Concept.
9	For Concepts inherited from published terminologies, the source code is retained in the concept_code field and can be used to reference the source vocabulary.

No.	Convention Description
10	Standard Concepts (designated as 'S' in the standard_concept field) may appear in CDM tables in all *_concept_id fields, whereas Classification Concepts ('C') should not appear in the CDM data, but participate in the construction of the CONCEPT_ANCESTOR table and can be used to identify Descendants that may appear in the data. See CONCEPT_ANCESTOR table. Non-standard Concepts can only appear in *_source_concept_id fields and are not used in CONCEPT_ANCESTOR table. Please refer to the Standardized Vocabularies specifications for details of the Standard Concept designation.
11	All logical data elements associated with the various CDM tables (usually in the _type_concept_id field) are called Type Concepts, including defining characteristics, qualifying attributes etc. They are also stored as Concepts in the CONCEPT table. Since they are generated by OMOP, their is no meaningful concept_code.
12	The lifespan of a Concept is recorded through its valid_start_date, valid_end_date and the invalid_reason fields. This allows Concepts to correctly reflect at which point in time were defined. Usually, Concepts get deprecated if their meaning was deemed ambiguous, a duplication of another Concept, or needed revision for scientific reason. For example, drug ingredients get updated when different salt or isomer variants enter the market. Usually, drugs taken off the market do not cause a deprecation by the terminology vendor. Since observational data are valid with respect to the time they are recorded, it is key for the Standardized Vocabularies to provide even obsolete codes and maintain their relationships to other current Concepts.
13	Concepts without a known instantiated date are assigned valid_start_date of '1-Jan-1970'.
14	Concepts that are not invalid are assigned valid_end_date of '31-Dec-2099'.
15	Deprecated Concepts (with a valid_end_date before the release date of the Standardized Vocabularies) will have a value of 'D' (deprecated without successor) or 'U' (updated). The updated Concepts have a record in the CONCEPT_RELATIONSHIP table indicating their active replacement Concept.
16	Values for CONCEPT_IDs generated as part of Standardized Vocabularies will be reserved from 0 to 2,000,000,000. Above this range, CONCEPT_IDs are available for local use and are guaranteed not to clash with future releases of the Standardized Vocabularies.

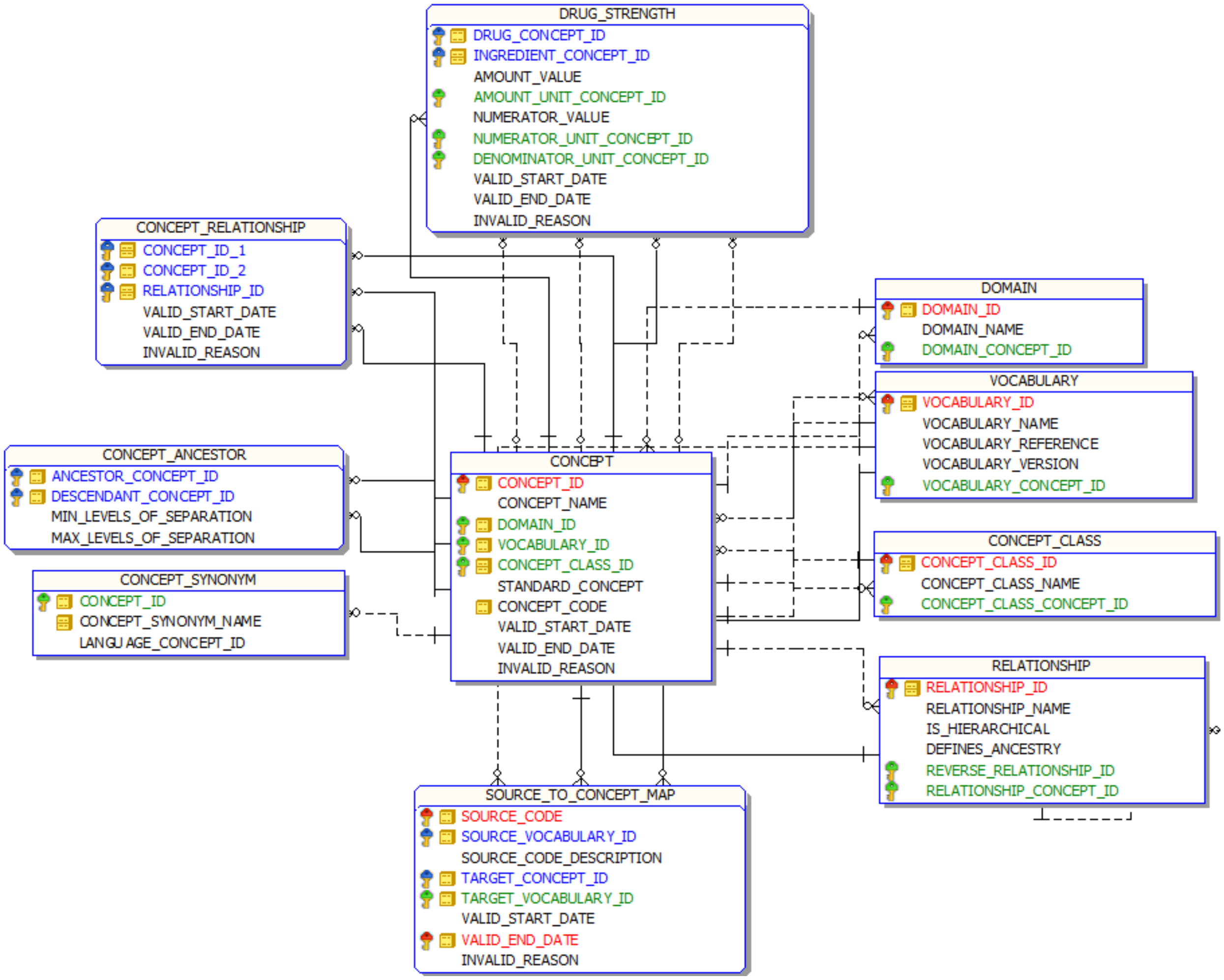


Figure 1: Vocabulary entity-relationship diagram

4.2 VOCABULARY

The VOCABULARY table includes a list of the Vocabularies collected from various sources or created de novo by the OMOP community. This reference table is populated with a single record for each Vocabulary source and includes a descriptive name and other associated attributes for the Vocabulary.

Field	Required	Type	Description
vocabulary_id	Yes	varchar(20)	A unique identifier for each Vocabulary, such as ICD9CM, SNOMED, Visit.
vocabulary_name	Yes	varchar(255)	The name describing the vocabulary, for example “International Classification of Diseases, Ninth Revision, Clinical Modification, Volume 1 and 2 (NCHS)” etc.
vocabulary_reference	Yes	varchar(255)	External reference to documentation or available download of the about the vocabulary.
vocabulary_version	No	varchar(255)	Version of the Vocabulary as indicated in the source.
vocabulary_concept_id	Yes	integer	A foreign key that refers to a standard concept identifier in the CONCEPT table for the Vocabulary the VOCABULARY record belongs to.

4.2.1 Conventions

No.	Convention Description
1	There is one record for each Vocabulary. One Vocabulary source or vendor can issue several Vocabularies, each of them creating their own record in the VOCABULARY table. However, the choice of whether a Vocabulary contains Concepts of different Concept Classes, or when these different classes constitute separate Vocabularies cannot precisely be decided based on the definition of what constitutes a Vocabulary. For example, the ICD-9 Volume 1 and 2 codes (ICD9CM, containing predominantly conditions and some procedures and observations) and the ICD-9 Volume 3 codes (ICD9Proc, containing predominantly procedures) are realized as two different Vocabularies. On the other hand, SNOMED-CT codes of the class Condition and those of the class Procedure are part of one and the same Vocabulary. Please refer to the Standardized Vocabularies specifications for details of each Vocabulary.
2	The VOCABULARY_ID field contains an alphanumerical identifier, that can also be used as the abbreviation of the Vocabulary name.
3	The record with VOCABULARY_ID = ‘None’ is reserved to contain information regarding the current version of the Entire Standardized Vocabularies.

No.	Convention Description
4	The VOCABULARY_NAME field contains the full official name of the Vocabulary, as well as the source or vendor in parenthesis.
5	Each Vocabulary has an entry in the CONCEPT table, which is recorded in the VOCABULARY_CONCEPT_ID field. This is for purposes of creating a closed Information Model, where all entities in the OMOP CDM are covered by a unique Concept.

4.3 DOMAIN

The DOMAIN table includes a list of OMOP-defined Domains the Concepts of the Standardized Vocabularies can belong to. A Domain defines the set of allowable Concepts for the standardized fields in the CDM tables. For example, the "Condition" Domain contains Concepts that describe a condition of a patient, and these Concepts can only be stored in the condition_concept_id field of the CONDITION_OCCURRENCE and CONDITION_ERA tables. This reference table is populated with a single record for each Domain and includes a descriptive name for the Domain.

Field	Required	Type	Description
domain_id	Yes	varchar(20)	A unique key for each domain.
domain_name	Yes	varchar(255)	The name describing the Domain, e.g. "Condition", "Procedure", "Measurement" etc.
domain_concept_id	Yes	integer	A foreign key that refers to an identifier in the CONCEPT table for the unique Domain Concept the Domain record belongs to.

4.3.1 Conventions

No.	Convention Description
1	There is one record for each Domain. The domains are defined by the tables and fields in the OMOP CDM that can contain Concepts describing all the various aspects of the healthcare experience of a patient.
2	The DOMAIN_ID field contains an alphanumerical identifier, that can also be used as the abbreviation of the Domain.
3	The DOMAIN_NAME field contains the unabbreviated names of the Domain.
4	Each Domain also has an entry in the Concept table, which is recorded in the DOMAIN_CONCEPT_ID field. This is for purposes of creating a closed Information Model, where all entities in the OMOP CDM are covered by unique Concepts.
5	Versions prior to v5.0.0 of the OMOP CDM did not support the notion of a Domain.

No.	Convention Description

4.4 CONCEPT_CLASS

The CONCEPT_CLASS table is a reference table, which includes a list of the classifications used to differentiate Concepts within a given Vocabulary. This reference table is populated with a single record for each Concept Class:

Field	Required	Type	Description
concept_class_id	Yes	varchar(20)	A unique key for each class.
concept_class_name	Yes	varchar(255)	The name describing the Concept Class, e.g. "Clinical Finding", "Ingredient", etc.
concept_class_concept_id	Yes	integer	A foreign key that refers to an identifier in the CONCEPT table for the unique Concept Class the record belongs to.

4.4.1 Conventions

No.	Convention Description
1	There is one record for each Concept Class. Concept Classes are used to create additional structure to the Concepts within each Vocabulary. Some Concept Classes are unique to a Vocabulary (for example 'Clinical Finding' in SNOMED), but others can be used across different Vocabularies. The separation of Concepts through Concept Classes can be semantically horizontal (each Class subsumes Concepts of the same hierarchical level, akin to sub-Vocabularies within a Vocabulary) or vertical (each Class subsumes Concepts of a certain kind, going across hierarchical levels). For example, Concept Classes in SNOMED are vertical: The classes "Procedure" and "Clinical Finding" define very granular to very generic Concepts. On the other hand, 'Clinical Drug' and 'Ingredient' Concept Classes define horizontal layers or strata in the RxNorm vocabulary, which all belong to the same concept of a Drug.
2	The CONCEPT_CLASS_ID field contains an alphanumerical identifier, that can also be used as the abbreviation of the Concept Class.
3	The CONCEPT_CLASS_NAME field contains the unabbreviated names of the Concept Class.
4	Each Concept Class also has an entry in the Concept table, which is recorded in the concept_class_concept_id field. This is for purposes of creating a closed Information Model, where all entities in the OMOP CDM are covered by unique Concepts.

4.5 CONCEPT_RELATIONSHIP

The CONCEPT_RELATIONSHIP table contains records that define direct relationships between any two Concepts and the nature or type of the relationship. Each type of a relationship is defined in the RELATIONSHIP table.

Field	Required	Type	Description
concept_id_1	Yes	integer	A foreign key to a Concept in the CONCEPT table associated with the relationship. Relationships are directional, and this field represents the source concept designation.
concept_id_2	Yes	integer	A foreign key to a Concept in the CONCEPT table associated with the relationship. Relationships are directional, and this field represents the destination concept designation.
relationship_id	Yes	varchar(20)	A unique identifier to the type or nature of the Relationship as defined in the RELATIONSHIP table.
valid_start_date	Yes	date	The date when the instance of the Concept Relationship is first recorded.
valid_end_date	Yes	date	The date when the Concept Relationship became invalid because it was deleted or superseded (updated) by a new relationship. Default value is 31-Dec-2099.
invalid_reason	No	varchar(1)	Reason the relationship was invalidated. Possible values are 'D' (deleted), 'U' (replaced with an update) or NULL when valid_end_date has the default value.

4.5.1 Conventions

No.	Convention Description
1	Relationships can generally be classified as hierarchical (parent-child) or non-hierarchical (lateral).
2	All Relationships are directional, and each Concept Relationship is represented twice symmetrically within the CONCEPT_RELATIONSHIP table. For example, the two SNOMED concepts of 'Acute myocardial infarction of the anterior wall' and 'Acute myocardial infarction' have two Concept Relationships: 1- 'Acute myocardial infarction of the anterior wall' 'Is a' 'Acute myocardial infarction', and 2- 'Acute myocardial infarction' 'Subsumes' 'Acute myocardial infarction of the anterior wall'.
3	There is one record for each Concept Relationship connecting the same Concepts with the same RELATIONSHIP_ID.

No.	Convention Description
4	Since all Concept Relationships exist with their mirror image (concept_id_1 and concept_id_2 swapped, and the RELATIONSHIP_ID replaced by the REVERSE_RELATIONSHIP_ID from the RELATIONSHIP table), it is not necessary to query for the existence of a relationship both in the concept_id_1 and concept_id_2 fields.
5	Concept Relationships define direct relationships between Concepts. Indirect relationships through 3rd Concepts are not captured in this table. However, the CONCEPT_ANCESTOR table does this for hierachical relationships over several "generations" of direct relationships.

4.6 RELATIONSHIP

The RELATIONSHIP table provides a reference list of all types of relationships that can be used to associate any two concepts in the CONCEPT_RELATIONSHP table.

Field	Required	Type	Description
relationship_id	Yes	varchar(20)	The type of relationship captured by the relationship record.
relationship_name	Yes	varchar(255)	The text that describes the relationship type.
is_hierarchical	Yes	varchar(1)	Defines whether a relationship defines concepts into classes or hierarchies. Values are 1 for hierarchical relationship or 0 if not.
defines_ancestry	Yes	varchar(1)	Defines whether a hierarchical relationship contributes to the concept_ancestor table. These are subsets of the hierarchical relationships. Valid values are 1 or 0.
reverse_relationship_id	Yes	varchar(20)	The identifier for the relationship used to define the reverse relationship between two concepts.
relationship_concept_id	Yes	integer	A foreign key that refers to an identifier in the CONCEPT table for the unique relationship concept.

4.6.1 Conventions

No.	Convention Description
1	There is one record for each Relationship.
2	Relationships are classified as hierarchical (parent-child) or non-hierarchical (lateral)
3	They are used to determine which concept relationship records should be included in the computation of the CONCEPT_ANCESTOR table.

No.	Convention Description
4	The RELATIONSHIP_ID field contains an alphanumerical identifier, that can also be used as the abbreviation of the Relationship.
5	The RELATIONSHIP_NAME field contains the unabbreviated names of the Relationship.
6	Relationships all exist symmetrically, i.e. in both direction. The RELATIONSHIP_ID of the opposite Relationship is provided in the REVERSE_RELATIONSHIP_ID field.
7	Each Relationship also has an equivalent entry in the Concept table, which is recorded in the RELATIONSHIP_CONCEPT_ID field. This is for purposes of creating a closed Information Model, where all entities in the OMOP CDM are covered by unique Concepts.
8	Hierarchical Relationships are used to build a hierarchical tree out of the Concepts, which is recorded in the CONCEPT_ANCESTOR table. For example, 'has_ingredient' is a Relationship between Concept of the Concept Class 'Clinical Drug' and those of 'Ingredient', and all Ingredients can be classified as the 'parental' hierarchical Concepts for the drug products they are part of. All 'Is a' Relationships are hierarchical.
9	Relationships, also hierarchical, can be between Concepts within the same Vocabulary or those adopted from different Vocabulary sources.

4.7 CONCEPT_SYNONYM

The CONCEPT_SYNONYM table is used to store alternate names and descriptions for Concepts.

Field	Required	Type	Description
concept_id	Yes	Integer	A foreign key to the Concept in the CONCEPT table.
concept_synonym_name	Yes	varchar(1000)	The alternative name for the Concept.
language_concept_id	Yes	integer	A foreign key to a Concept representing the language.

4.7.1 Conventions

No.	Convention Description
1	The concept_synonym_name field contains a valid Synonym of a concept, including the description in the concept_name itself. i.e., each Concept has at least one Synonym in the CONCEPT_SYNONYM table. As an example, for a SNOMED-CT Concept, if the fully specified name is stored as the concept_name of the CONCEPT table, then the Preferred Term and Synonyms associated with the Concept are stored in the CONCEPT_SYNONYM table.
2	Only Synonyms that are active and current are stored in the CONCEPT_SYNONYM table. Tracking synonym/description history and mapping of obsolete synonyms to current Concepts/Synonyms is out of scope for the Standard Vocabularies.
3	Currently, only English Synonyms are included.

4.8 CONCEPT_ANCESTOR

The CONCEPT_ANCESTOR table is designed to simplify observational analysis by providing the complete hierarchical relationships between Concepts. Only direct parent-child relationships between Concepts are stored in the CONCEPT_RELATIONSHIP table. To determine higher level ancestry connections, all individual direct relationships would have to be navigated at analysis time. The CONCEPT_ANCESTOR table includes records for all parent-child relationships, as well as grandparent-grandchild relationships and those of any other level of lineage. Using the CONCEPT_ANCESTOR table allows for querying for all descendants of a hierarchical concept. For example, drug ingredients and drug products are all descendants of a drug class ancestor.

This table is entirely derived from the CONCEPT, CONCEPT_RELATIONSHIP and RELATIONSHIP tables.

Field	Required	Type	Description
ancestor_concept_id	Yes	integer	A foreign key to the concept in the concept table for the higher-level concept that forms the ancestor in the relationship.
descendant_concept_id	Yes	integer	A foreign key to the concept in the concept table for the lower-level concept that forms the descendant in the relationship.
min_levels_of_separation	Yes	integer	The minimum separation in number of levels of hierarchy between ancestor and descendant concepts. This is an attribute that is used to simplify hierarchic analysis.
max_levels_of_separation	Yes	integer	The maximum separation in number of levels of hierarchy between ancestor and descendant concepts. This is an attribute that is used to simplify hierarchic analysis.

4.8.1 Conventions

No.	Convention Description
1	Each concept is also recorded as an ancestor of itself.
2	Only valid and Standard Concepts participate in the CONCEPT_ANCESTOR table. It is not possible to find ancestors or descendants of deprecated or Source Concepts.
3	Usually, only Concepts of the same Domain are connected through records of the CONCEPT_ANCESTOR table, but there might be exceptions.

4.9 SOURCE_TO_CONCEPT_MAP

The source to concept map table is a legacy data structure within the OMOP Common Data Model, recommended for use in ETL processes to maintain local source codes which are not available as Concepts in the Standardized Vocabularies, and to establish mappings for each source code into a Standard Concept as target_concept_ids that can be used to populate the Common Data Model tables. The SOURCE_TO_CONCEPT_MAP table is no longer populated with content within the Standardized Vocabularies published to the OMOP community.

Field	Required	Type	Description
source_code	Yes	varchar(50)	The source code being translated into a Standard Concept.
source_concept_id	Yes	integer	A foreign key to the Source Concept that is being translated into a Standard Concept.
source_vocabulary_id	Yes	varchar(20)	A foreign key to the VOCABULARY table defining the vocabulary of the source code that is being translated to a Standard Concept.
source_code_description	No	varchar(255)	An optional description for the source code. This is included as a convenience to compare the description of the source code to the name of the concept.
target_concept_id	Yes	integer	A foreign key to the target Concept to which the source code is being mapped.
target_vocabulary_id	Yes	varchar(20)	A foreign key to the VOCABULARY table defining the vocabulary of the target Concept.
valid_start_date	Yes	date	The date when the mapping instance was first recorded.
valid_end_date	Yes	date	The date when the mapping instance became invalid because it was deleted or superseded (updated) by a new relationship. Default value is 31-Dec-2099.

Field	Required	Type	Description
invalid_reason	No	varchar(1)	Reason the mapping instance was invalidated. Possible values are D (deleted), U (replaced with an update) or NULL when valid_end_date has the default value.

4.9.1 Conventions

No.	Convention Description
1	This table is no longer used to distribute mapping information between source codes and Standard Concepts for the Standard Vocabularies. Instead, the CONCEPT_RELATIONSHIP table is used for this purpose, using the relationship_id='Maps to'.
2	However, this table can still be used for the translation of local source codes into Standard Concepts.
4	**Note:** This table should not be used to translate source codes to Source Concepts. The source code of a Source Concept is captured in its concept_code field. If the source codes used in a given database do not follow correct formatting the ETL will have to perform this translation. For example, if ICD-9-CM codes are recorded without a dot the ETL will have to perform a lookup function that allows identifying the correct ICD-9-CM Source Concept (with the dot in the concept_code field).
5	The SOURCE_CONCEPT_ID, or the combination of the fields source_code and the SOURCE_VOCABULARY_ID uniquely identifies the source information. It is the equivalent to the CONCEPT_ID_1 field in the CONCEPT_RELATIONSHIP table.
6	If there is no SOURCE_CONCEPT_ID available because the source codes are local and not supported by the Standard Vocabulary, the content of the field is 0 (zero, not null) encoding an undefined concept. However, local Source Concepts are established (concept_id values above 2,000,000,000).
7	The SOURCE_CODE_DESCRIPTION contains an optional description of the source code.
8	The TARGET_CONCEPT_ID contains the Concept the source code is mapped to. It is equivalent to the concept_id_2 in the CONCEPT_RELATIONSHIP table
9	The TARGET_VOCABULARY_ID field contains the VOCABULARY_ID of the target concept. It is a duplication of the same information in the CONCEPT record of the Target Concept.

No.	Convention Description
10	The fields VALID_START_DATE, VALID_END_DATE and INVALID_REASON are used to define the life cycle of the mapping information. Invalid mapping records should not be used for mapping information.

4.10 DRUG_STRENGTH

The DRUG_STRENGTH table contains structured content about the amount or concentration and associated units of a specific ingredient contained within a particular drug product. This table is supplemental information to support standardized analysis of drug utilization.

Field	Required	Type	Description
drug_concept_id	Yes	integer	A foreign key to the Concept in the CONCEPT table representing the identifier for Branded Drug or Clinical Drug Concept.
ingredient_concept_id	Yes	integer	A foreign key to the Concept in the CONCEPT table, representing the identifier for drug Ingredient Concept contained within the drug product.
amount_value	No	float	The numeric value associated with the amount of active ingredient contained within the product.
amount_unit_concept_id	No	integer	A foreign key to the Concept in the CONCEPT table representing the identifier for the Unit for the absolute amount of active ingredient.
numerator_value	No	float	The numeric value associated with the concentration of the active ingredient contained in the product
numerator_unit_concept_id	No	integer	A foreign key to the Concept in the CONCEPT table representing the identifier for the numerator Unit for the concentration of active ingredient.
denominator_value	No	float	The amount of total liquid (or other divisible product, such as ointment, gel, spray, etc.).
denominator_unit_concept_id	No	integer	A foreign key to the Concept in the CONCEPT table representing the identifier for the denominator Unit for the concentration of active ingredient.
box_size	No	integer	The number of units of Clinical of Branded Drug, or Quantified Clinical or Branded Drug contained in a box as dispensed to the patient
valid_start_date	Yes	date	The date when the Concept was first recorded. The default value is 1-Jan-1970.

Field	Required	Type	Description
valid_end_date	Yes	date	The date when the concept became invalid because it was deleted or superseded (updated) by a new Concept. The default value is 31-Dec-2099.
invalid_reason	No	varchar(1)	Reason the concept was invalidated. Possible values are 'D' (deleted), 'U' (replaced with an update) or NULL when valid_end_date has the default value.

4.10.1 Conventions

No.	Convention Description
1	The DRUG_STRENGTH table contains information for each active (non-deprecated) standard drug concept.
2	A drug which contains multiple active Ingredients will result in multiple DRUG_STRENGTH records, one for each active ingredient.
3	Ingredient strength information is provided either as absolute amount (usually for solid formulations) or as concentration (usually for liquid formulations).
4	If the absolute amount is provided (for example, 'Acetaminophen 5 MG Tablet') the AMOUNT_VALUE and AMOUNT_UNIT_CONCEPT_ID are used to define this content (in this case 5 and 'MG').
5	If the concentration is provided (for example 'Acetaminophen 48 MG/ML Oral Solution') the NUMERATOR_VALUE in combination with the NUMERATOR_UNIT_CONCEPT_ID and DENOMINATOR_UNIT_CONCEPT_ID are used to define this content (in this case 48, 'MG' and 'ML').
6	In case of Quantified Clinical or Branded Drugs the DENOMINATOR_VALUE contains the total amount of the solution (not the amount of the ingredient). In all other drug concept classes the denominator amount is NULL because the concentration is always normalized to the unit of the denominator. So, a product containing 960 mg in 20 mL is provided as 48 mg/mL in the Clinical Drug and Clinical Drug Component, while as a Quantified Clinical Drug it is written as 960 mg/20 mL.

No.	Convention Description
7	If the strength is provided in % (volume or mass-percent are not distinguished) it is stored in the NUMERATOR_VALUE/NUMERATOR_UNIT_CONCEPT_ID field combination, with both the DENOMINATOR_VALUE and DENOMINATOR_UNIT_CONCEPT_ID set to NULL. If it is a Quantified Drug the total amount of drug is provided in the DENOMINATOR_VALUE/DENOMINATOR_UNIT_CONCEPT_ID pair. E.g., the 30 G Isoconazole 2% Topical Cream is provided as 2% / in Clinical Drug and Clinical Drug Component, and as 2% /30 G.
8	Sometimes, one Ingredient is listed with different units within the same drug. This is very rare, and usually this happens if there are more than one Precise Ingredient. For example, 'Penicillin G, Benzathine 150000 UNT/ML / Penicillin G, Procaine 150000 MEQ/ML Injectable Suspension' contains Penicillin G in two different forms.
9	Sometimes, different ingredients in liquid drugs are listed with different units in the DENOMINATOR_UNIT_CONCEPT_ID. This is usually the case if the ingredients are liquids themselves (concentration provided as mL/mL) or solid substances (mg/mg). In these cases, the general assumptions is made that the density of the drug is that of water, and one can assume 1 g = 1 mL.
10	All Drug vocabularies containing Standard Concepts have entries in the DRUG_STRENGTH table.
11	There is now a Concept Class for supplier information whose relationships can be found in CONCEPT_RELATIONSHIP with a RELATIONSHIP_ID of 'Has supplier' and 'Supplier of'

5 Standardized Metadata

CDM_SOURCE METADATA

All metadata about the data should be derived from the data themselves.

5.1 CDM_SOURCE

The CDM_SOURCE table contains detail about the source database and the process used to transform the data into the OMOP Common Data Model.

Field	Required	Type	Description
cdm_source_name	Yes	varchar(255)	The full name of the source
cdm_source_abbreviation	No	varchar(25)	An abbreviation of the name

Field	Required	Type	Description
cdm_holder	No	varchar(255)	The name of the organization responsible for the development of the CDM instance
source_description	No	CLOB	A description of the source data origin and purpose for collection. The description may contain a summary of the period of time that is expected to be covered by this dataset.
source_documentation_reference	No	varchar(255)	URL or other external reference to location of source documentation
cdm_etl_reference	No	varchar(255)	URL or other external reference to location of ETL specification documentation and ETL source code
source_release_date	No	date	The date for which the source data are most current, such as the last day of data capture
cdm_release_date	No	date	The date when the CDM was instantiated
cdm_version	No	varchar(10)	The version of CDM used
vocabulary_version	No	varchar(20)	The version of the vocabulary used

5.1.1 Conventions

No.	Convention Description
1	If a source database is derived from multiple data feeds, the integration of those disparate sources is expected to be documented in the ETL specifications. The source information on each of the databases can be represented as separate records in the CDM_SOURCE table.
2	Currently, there is no mechanism to link individual records in the CDM tables to their source record in the CDM_SOURCE table.
3	The version of the vocabulary can be obtained from the vocabulary_name field in the VOCABULARY table for the record where vocabulary_id='None'.

5.2 METADATA

The METADATA table contains metadata information about a dataset that has been transformed to the OMOP Common Data Model.

Field	Required	Type	Description
metadata_concept_id	Yes	integer	A foreign key that refers to a Standard Metadata Concept identifier in the Standardized Vocabularies.
metadata_type_concept_id	Yes	integer	A foreign key that refers to a Standard Type Concept identifier in the Standardized Vocabularies.

Field	Required	Type	Description
name	Yes	varchar(250)	The name of the Concept stored in metadata_concept_id or a description of the data being stored.
value_as_string	No	nvarchar	The metadata value stored as a string.
value_as_concept_id	No	integer	A foreign key to a metadata value stored as a Concept ID.
metadata date	No	date	The date associated with the metadata
metadata_datetime	No	datetime	The date and time associated with the metadata

5.2.1 Conventions

No.	Convention Description
1	One record in the Metadata table is pre-populated in the DDL indicating the CDM version of the database.

6 Standardized Clinical Data Tables

PERSON
OBSERVATION_PERIOD
DEATH
VISIT_OCCURRENCE
VISIT_DETAIL CONDITION_OCCURRENCE DRUG_EXPOSURE
PROCEDURE_OCCURRENCE
DEVICE_EXPOSURE
MEASUREMENT
NOTE
NOTE_NLP SURVEY_CONDUCT
OBSERVATION SPECIMEN
FACT_RELATIONSHIP

These tables contain the core information about the clinical events that occurred longitudinally during valid Observation Periods for each Person, as well as demographic information for the Person. Below provides an entity-relationship diagram highlighting the tables within the Standardized Clinical Data portion of the OMOP Common Data Model:

6.1 PERSON

The Person Domain contains records that uniquely identify each patient in the source data who is time at-risk to have clinical observations recorded within the source systems.

Field	Required	Type	Description
person_id	Yes	integer	A unique identifier for each person.
gender_concept_id	Yes	integer	A foreign key that refers to an identifier in the CONCEPT table for the unique gender of the person.

Field	Required	Type	Description
year_of_birth	Yes	integer	The year of birth of the person. For data sources with date of birth, the year is extracted. For data sources where the year of birth is not available, the approximate year of birth is derived based on any age group categorization available.
month_of_birth	No	integer	The month of birth of the person. For data sources that provide the precise date of birth, the month is extracted and stored in this field.
day_of_birth	No	integer	The day of the month of birth of the person. For data sources that provide the precise date of birth, the day is extracted and stored in this field.
birth_datetime	No	datetime	The date and time of birth of the person.
death_datetime	No	datetime	The date and time of death of the person.
race_concept_id	Yes	integer	A foreign key that refers to an identifier in the CONCEPT table for the unique race of the person, belonging to the 'Race' vocabulary.
ethnicity_concept_id	Yes	integer	A foreign key that refers to the standard concept identifier in the Standardized Vocabularies for the ethnicity of the person, belonging to the 'Ethnicity' vocabulary.
location_id	No	integer	A foreign key to the place of residency for the person in the location table, where the detailed address information is stored.
provider_id	No	integer	A foreign key to the primary care provider the person is seeing in the provider table.
care_site_id	No	integer	A foreign key to the site of primary care in the care_site table, where the details of the care site are stored.
person_source_value	No	varchar(50)	An (encrypted) key derived from the person identifier in the source data. This is necessary when a use case requires a link back to the person data at the source dataset.
gender_source_value	No	varchar(50)	The source code for the gender of the person as it appears in the source data. The personâ€™s gender is mapped to a standard gender concept in the Standardized Vocabularies; the original value is stored here for reference.
gender_source_concept_id	Yes	Integer	A foreign key to the gender concept that refers to the code used in the source.
race_source_value	No	varchar(50)	The source code for the race of the person as it appears in the source data. The person race is mapped to a standard race concept in the Standardized Vocabularies and the original value is stored here for reference.
race_source_concept_id	Yes	Integer	A foreign key to the race concept that refers to the code used in the source.

Field	Required	Type	Description
ethnicity_source_value	No	varchar(50)	The source code for the ethnicity of the person as it appears in the source data. The person ethnicity is mapped to a standard ethnicity concept in the Standardized Vocabularies and the original code is, stored here for reference.
ethnicity_source_concept_id	Yes	Integer	A foreign key to the ethnicity concept that refers to the code used in the source.

6.1.1 Conventions

No.	Convention Description
1	All tables representing patient-related Domains have a foreign-key reference to the person_id field in the PERSON table.
2	Each person record has associated demographic attributes which are assumed to be constant for the patient throughout the course of their periods of observation. For example, the location or gender is expected to have a unique value per person, even though in life these data may change over time.
3	The GENDER_CONCEPT_ID should store what is believed to be the biological or sex assigned at birth. If the data set does have gender identification information, this should be stored in the OBSERVATION table (using the gender concepts 8532-Female or 8507-Male in OBSERVATION_CONCEPT_ID)THEMIS issue #32.
4	If we do not know the month or day of birth, we do not guess. A person can exist without a month or day of birth. If a person lacks a birth year that person should be dropped(THEMIS issue #30).
5	Living patients should not have a value in PERSON.DEATH_DATETIME, nor should they have any records relating to death either in the CONDITION_OCCURRENCE or OBSERVATION tables
6	Only one death date per individual can be used. If a patient has clinical activity (e.g. prescriptions filled, labs performed, etc) more than 60+ days after death you may want to drop the death record as it may have been falsely reported. If multiple records of death exist on multiple days you may select the death that you deem most reliable (e.g. death at discharge) or select the latest death date.
7	If multiple death records occur, the date and the person have to be the same, but the cause can be different. Can be reported by different sources as well.

No.	Convention Description
8	If PERSON.DEATH_DATETIME cannot be precisely determined from the data, the best approximation should be used.

| 9 | The DEATH_DATETIME in the PERSON table should not be used as the way to find all deaths

`select * from PERSON where death_datetime is not null` should not be the practice

Rather, deaths should be found through the OBSERVATION table and the PERSON table is only used to determine which death date should be used in analysis

10 | Valid Gender, Race and Ethnicity Concepts each belong to their own Domain.
11 | Ethnicity in the OMOP CDM follows the OMB Standards for Data on Race and Ethnicity: Only distinctions between Hispanics and Non-Hispanics are made.
12 | Additional information is stored through references to other tables, such as the home address (location_id) or the primary care provider.
13 | The Provider refers to the primary care provider (General Practitioner). When the primary provider is unknown for a person then leave the PROVIDER_ID blank (THEMIS issue #36).
14 | The Care Site refers to where the Provider typically provides the primary care. When care site for the primary provider is unknown then leave the CARE_SITE_ID blank.
15 | It is not required that all subjects from the raw data be carried over to the CDM, in fact removing people that are not of high enough quality may help researchers using the CDM. Example scenarios to remove subjects include: a personâ€™s year of birth or age are unreasonable (e.g. born in year 0, 1800, 2999 or just lacking a year of birth), person lacks health benefits in claims database (i.e. thus you do not have a complete picture of their record), or raw data states that the person may not be of high research quality (e.g. CPRD will actually suggest which people not to use within research). Removal of a patient is not required and should be made in consideration of the raw data source. Reasons for removal of persons should be documented in the ETL documentation and METADATA table (insert row in METADATA where metadata.name='count of removed persons' and metada.value_as_string='xyz' where xyz is a number (e.g., 12). An ETL should not delete persons who contribute time however have no health care utilization (e.g. an individual enrolled in insurance but does not visit a doctor or pharmacy). This individual will contribute to analysis however as a healthy / non-care seeking individual (THEMIS issue #9).|

6.2 OBSERVATION_PERIOD

The OBSERVATION_PERIOD table contains records which uniquely define the spans of time for which a Person is at-risk to have clinical events recorded within the source systems, even if no events in fact are recorded (healthy patient with no healthcare interactions).

Field	Required	Type	Description
observation_period_id	Yes	integer	A unique identifier for each observation period.
person_id	Yes	integer	A foreign key identifier to the person for whom the observation period is defined. The demographic details of that person are stored in the person table.
observation_period_start_date	Yes	date	The start date of the observation period for which data are available from the data source.
observation_period_end_date	Yes	date	The end date of the observation period for which data are available from the data source.

Field	Required	Type	Description
period_type_concept_id	Yes	Integer	A foreign key identifier to the predefined concept in the Standardized Vocabularies reflecting the source of the observation period information, belonging to the 'Obs Period Type' vocabulary

6.2.1 Conventions

No.	Convention Description
1	Each Person has to have at least one observation period.
2	One Person may have one or more disjoint observation periods, during which times analyses may assume that clinical events would be captured if observed
3	Each Person can have more than one valid OBSERVATION_PERIOD record, but no two observation periods can overlap in time for a given person.
4	As a general assumption, during an Observation Period any clinical event that happens to the patient is expected to be recorded. Conversely, the absence of data indicates that no clinical events occurred to the patient.
5	Both the _START_DATE and the _END_DATE of the clinical event has to be between observation_period_start_date and observation_period_end_date.
6	Events CAN fall outside of an observation period though they should fall in a valid payer plan period, such as Medicare Part D, which can overlap an observation period. However, time outside of an observation period cannot be used to identify people. To ensure quality, events outside of an observation period should not be used for analysis. THEMIS issue #23
7	For claims data, observation periods are inferred from the enrollment periods to a health benefit plan.
8	For EHR data, the observation period cannot be determined explicitly, because patients usually do not announce their departure from a certain healthcare provider. The ETL will have to apply some heuristic to make a reasonable guess on what the observation_period should be. Refer to the ETL documentation for details.

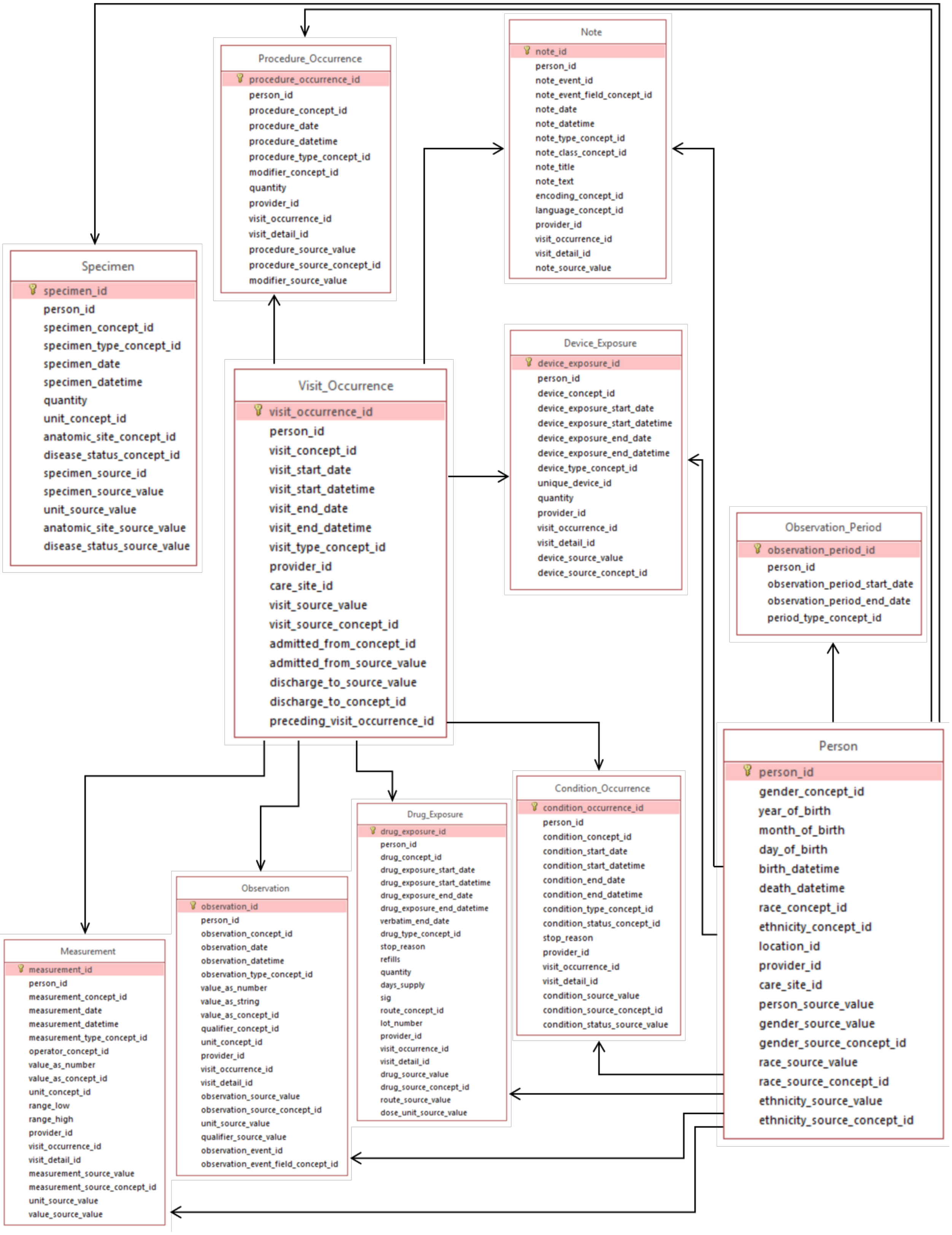

Procedure_Occurrence
procedure_occurrence_id
person_id
procedure_concept_id
procedure_date
procedure_datetime
procedure_type_concept_id
modifier_concept_id
quantity
provider_id
visit_occurrence_id
visit_detail_id
procedure_source_value
procedure_source_concept_id
modifier_source_value
Note
note_id
person_id
note_event_id
note_event_field_concept_id
note_date
note_datetime
note_type_concept_id
note_class_concept_id
note_title
note_text
encoding_concept_id
language_concept_id
provider_id
visit_occurrence_id
visit_detail_id
note_source_value
Specimen
specimen_id
person_id
specimen_concept_id
specimen_type_concept_id
specimen_date
specimen_datetime
quantity
unit_concept_id
anatomic_site_concept_id
disease_status_concept_id
specimen_source_id
specimen_source_value
unit_source_value
anatomic_site_source_value
disease_status_source_value
Visit_Occurrence
visit_occurrence_id
person_id
visit_concept_id
visit_start_date
visit_start_datetime
visit_end_date
visit_end_datetime
visit_type_concept_id
provider_id
care_site_id
visit_source_value
visit_source_concept_id
admitted_from_concept_id
admitted_from_source_value
discharge_to_source_value
discharge_to_concept_id
preceding_visit_occurrence_id
Device_Exposure
device_exposure_id
person_id
device_concept_id
device_exposure_start_date
device_exposure_start_datetime
device_exposure_end_date
device_exposure_end_datetime
device_type_concept_id
unique_device_id
quantity
provider_id
visit_occurrence_id
visit_detail_id
device_source_value
device_source_concept_id
Observation_Period
observation_period_id
person_id
observation_period_start_date
observation_period_end_date
period_type_concept_id
Person
person_id
gender_concept_id
year_of_birth
month_of_birth
day_of_birth
birth_datetime
death_datetime
race_concept_id
ethnicity_concept_id
location_id
provider_id
care_site_id
person_source_value
gender_source_value
gender_source_concept_id
race_source_value
race_source_concept_id
ethnicity_source_value
ethnicity_source_concept_id
Condition_Occurrence
condition_occurrence_id
person_id
condition_concept_id
condition_start_date
condition_start_datetime
condition_end_date
condition_end_datetime
condition_type_concept_id
condition_status_concept_id
stop_reason
provider_id
visit_occurrence_id
visit_detail_id
condition_source_value
condition_source_concept_id
condition_status_source_value
Drug_Exposure
drug_exposure_id
person_id
drug_concept_id
drug_exposure_start_date
drug_exposure_start_datetime
drug_exposure_end_date
drug_exposure_end_datetime
verbatim_end_date
drug_type_concept_id
stop_reason
refills
quantity
days_supply
sig
route_concept_id
lot_number
provider_id
visit_occurrence_id
visit_detail_id
drug_source_value
drug_source_concept_id
route_source_value
dose_unit_source_value
Observation
observation_id
person_id
observation_concept_id
observation_date
observation_datetime
observation_type_concept_id
value_as_number
value_as_string
value_as_concept_id
qualifier_concept_id
unit_concept_id
provider_id
visit_occurrence_id
visit_detail_id
observation_source_value
observation_source_concept_id
unit_source_value
qualifier_source_value
observation_event_id
observation_event_field_concept_id
Measurement
measurement_id
person_id
measurement_concept_id
measurement_date
measurement_datetime
measurement_type_concept_id
operator_concept_id
value_as_number
value_as_concept_id
unit_concept_id
range_low
range_high
provider_id
visit_occurrence_id
visit_detail_id
measurement_source_value
measurement_source_concept_id
unit_source_value
value_source_value

Figure 2:

6.3 VISIT_OCCURRENCE

The VISIT_OCCURRENCE table contains the spans of time a Person continuously receives medical services from one or more providers at a Care Site in a given setting within the health care system. Visits are classified into 4 settings: outpatient care, inpatient confinement, emergency room, and long-term care. Persons may transition between these settings over the course of an episode of care (for example, treatment of a disease onset).

Field	Required	Type	Description
visit_occurrence_id	Yes	integer	A unique identifier for each Person's visit or encounter at a healthcare provider.
person_id	Yes	integer	A foreign key identifier to the Person for whom the visit is recorded. The demographic details of that Person are stored in the PERSON table.
visit_concept_id	Yes	integer	A foreign key that refers to a visit Concept identifier in the Standardized Vocabularies belonging to the 'Visit' Vocabulary.
visit_start_date	No	date	The start date of the visit.
visit_start_datetime	Yes	datetime	The date and time of the visit started.
visit_end_date	No	date	The end date of the visit. If this is a one-day visit the end date should match the start date.
visit_end_datetime	Yes	datetime	The date and time of the visit end.
visit_type_concept_id	Yes	Integer	A foreign key to the predefined Concept identifier in the Standardized Vocabularies reflecting the type of source data from which the visit record is derived belonging to the 'Visit Type' vocabulary.
provider_id	No	integer	A foreign key to the provider in the provider table who was associated with the visit.
care_site_id	No	integer	A foreign key to the care site in the care site table that was visited.
visit_source_value	No	varchar(50)	The source code for the visit as it appears in the source data.
visit_source_concept_id	Yes	integer	A foreign key to a Concept that refers to the code used in the source.
admitting_source_concept_id	Yes	integer	A foreign key to the predefined concept in the Place of Service Vocabulary reflecting the admitting source for a visit.
admitting_source_value	No	varchar(50)	The source code for the admitting source as it appears in the source data.
discharge_to_concept_id	Yes	integer	A foreign key to the predefined concept in the Place of Service Vocabulary reflecting the discharge disposition for a visit.
discharge_to_source_value	No	varchar(50)	The source code for the discharge disposition as it appears in the source data.
preceding_visit_occurrence_id	No	integer	A foreign key to the VISIT_OCCURRENCE table of the visit immediately preceding this visit

6.3.1 Conventions

No.	Convention Description
1	A Visit Occurrence is recorded for each visit to a healthcare facility.

No.	Convention Description
2	Valid Visit Concepts belong to the 'Visit' domain.
3	Standard Visit Concepts are defined, among others, as Inpatient Visit, Outpatient Visit, Emergency Room Visit, Long Term Care Visit and combined ER and Inpatient Visit. The latter is necessary because it is close to impossible to separate the two in many EHR system, treating them interchangeably. To annotate this correctly, the visit concept 'Emergency Room and Inpatient Visit' (concept_id=262) should be used.
4	Handling of death: In the case when a patient died during admission (VISIT_OCCURRENCE.DISCHARGE_TO_CONCEPT_ID = 4216643 'Patient died'), a record in the Observation table should be created with OBSERVATION_TYPE_CONCEPT_ID = 44818516 (EHR discharge status 'Expired').
5	Source Concepts from place of service vocabularies are mapped into these standard visit Concepts in the Standardized Vocabularies.
6	At any one day, there could be more than one visit.
7	One visit may involve multiple providers, in which case the ETL must specify how a single PROVIDER_ID is selected or leave the PROVIDER_ID field null.
8	One visit may involve multiple Care Sites, in which case the ETL must specify how a single CARE_SITE_ID is selected or leave the CARE_SITE_ID field null.

| 9 | Visits are recorded in various data sources in different forms with varying levels of standardization. For example:

Medical Claims include Inpatient Admissions, Outpatient Services, and Emergency Room visits.

Electronic Health Records may capture Person visits as part of the activities recorded depending whether the EHR system is used at the different Care Sites./li>

| | 10 | In addition to the 'Place of Service' vocabulary the following SNOMED concepts for discharge disposition (DISCHARGE_TO_CONCEPT_ID) can be used:

Patient died: 4216643

Absent without leave: 44814693

Patient self-discharge against medical advice: 4021968

| | 11 | PRECEDING_VISIT_ID can be used to link a visit immediately preceding the current visit. | | 12 | Visit end dates are mandatory. If end dates are not provided in the source there are three ways in which to derive them:

Outpatient Visit: VISIT_END_DATE = VISIT_START_DATE

Emergency Room Visit: VISIT_END_DATE = VISIT_START_DATE

Inpatient Visit: Usually there is information about discharge. If not, you should be able to derive the end date from the sudden decline of activity or from the absence of inpatient procedures/drugs.

Long Term Care Visits: Particularly for claims data, if end dates are not provided assume the visit is for the duration of month that it occurs.

For inpatient visits ongoing at the date of ETL, put date of processing the data as mandatory VISIT_END_DATE and VISIT_TYPE_CONCEPT_ID with 32220-“Still patient” to identify the visit as incomplete.

(THEMIS issue #13).|

6.4 VISIT_DETAIL

The VISIT_DETAIL table is an optional table used to represents details of each record in the parent visit_occurrence table. For every record in visit_occurrence table there may be 0 or more records in the visit_detail table with a 1:n relationship where n may be 0. The visit_detail table is structurally very similar to visit_occurrence table and belongs to the similar domain as the visit.

Field	Required	Type	Description
visit_detail_id	Yes	integer	A unique identifier for each Person’s visit or encounter at a healthcare provider.
person_id	Yes	integer	A foreign key identifier to the Person for whom the visit is recorded. The demographic details of that Person are stored in the PERSON table.
visit_detail_concept_id	Yes	integer	A foreign key that refers to a visit Concept identifier in the Standardized Vocabularies belonging to the ‘Visit’ Vocabulary.
visit_detail_start_date	No	date	The start date of the visit.
visit_detail_start_datetime	Yes	datetime	The date and time of the visit started.
visit_detail_end_date	No	date	The end date of the visit. If this is a one-day visit the end date should match the start date.
visit_detail_end_datetime	Yes	datetime	The date and time of the visit end.
visit_detail_type_concept_id	Yes	Integer	A foreign key to the predefined Concept identifier in the Standardized Vocabularies reflecting the type of source data from which the visit record is derived belonging to the ‘Visit Type’ vocabulary.
provider_id	No	integer	A foreign key to the provider in the provider table who was associated with the visit.
care_site_id	No	integer	A foreign key to the care site in the care site table that was visited.
visit_detail_source_value	No	string(50)	The source code for the visit as it appears in the source data.
visit_detail_source_concept_id	Yes	Integer	A foreign key to a Concept that refers to the code used in the source.
admitted_from_source_value	No	Varchar(50)	The source code for the admitting source as it appears in the source data.
admitted_from_concept_id	Yes	Integer	A foreign key to the predefined concept in the ‘Place of Service’ Vocabulary reflecting the admitting source for a visit.
discharge_to_source_value	No	Varchar(50)	The source code for the discharge disposition as it appears in the source data.
discharge_to_concept_id	Yes	Integer	A foreign key to the predefined concept in the ‘Place of Service’ Vocabulary reflecting the discharge disposition for a visit.
preceding_visit_detail_id	No	Integer	A foreign key to the VISIT_DETAIL table of the visit immediately preceding this visit

Field	Required	Type	Description
visit_detail_parent_id	No	Integer	A foreign key to the VISIT_DETAIL table record to represent the immediate parent visit-detail record.
visit_occurrence_id	Yes	Integer	A foreign key that refers to the record in the VISIT_OCCURRENCE table. This is a required field, because for every visit_detail is a child of visit_occurrence and cannot exist without a corresponding parent record in visit_occurrence.

6.4.1 Conventions

All conventions used in VISIT_OCCURRENCE apply to VISIT_DETAIL, with some notable exceptions as detailed below

No.	Convention Description
1	A Visit Detail is an optional detail record for each Visit Occurrence to a healthcare facility. For every record in VISIT_DETAIL there has to be a parent VISIT_OCCURRENCE record.
2	One record in VISIT_DETAIL can only have one VISIT_OCCURRENCE parent.
3	A single VISIT_OCCURRENCE record may have many child VISIT_DETAIL records.
4	Valid Visit Concepts belong to the 'Visit' domain. Standard Visit Concepts are yet to be defined, but will represent a detail of the Standard Visit Concept in VISIT_OCCURRENCE.
5	Handling of death: In the case when a patient died during admission (VISIT_DETAIL.DISCHARGE_TO_CONCEPT_ID = 4216643 'Patient died'), a record in the Observation table should be created with OBSERVATION_TYPE_CONCEPT_ID = 44818516 (EHR discharge status 'Expired').
6	Source Concepts from place of service vocabularies are mapped into these Standard Visit Concepts in the Standardized Vocabularies.
7	On any one day, there could be more than one visit. VISIT_OCCURRENCE allows for more than one visit within a single day. VISIT_DETAIL is to be used to only capture details within the visit.
8	One visit may involve multiple Providers, in which case, in VISIT_OCCURRENCE, the ETL must specify how a single PROVIDER_ID is selected or leave the PROVIDER_ID field null. VISIT_DETAIL allows for the ETL to specify multiple child records per VISIT_OCCURRENCE - and each of these child records may represent different PROVIDER_IDs.

No.	Convention Description
9	One visit may involve multiple Care Sites, in which case, in VISIT_OCCURRENCE, the ETL must specify how a single CARE_SITE_ID is selected or leave the CARE_SITE_ID field null. VISIT_DETAIL allows for the ETL to specify multiple child records per visit occurrence - and each of these child records may represent different CARE_SITEs.
10	Just like in VISIT_OCCURRENCE, records in VISIT_DETAIL may be sequentially related to each. These sequential relations are represented using PRECEDING_VISIT_DETAIL_ID.
11	Unlike VISIT_OCCURRENCE, VISIT_DETAIL may have nested visits with hierarchical relationships to each other. These relationships are represented using VISIT_DETAIL_PARENT_ID.
12	In US claims data Header/summary data that summarizes the entire claim and Line/detail that details a claim, detail is thus a child of the summary, and for every record in summary there is one or more records in detail. i.e. there will be at least one foreign key link from VISIT_DETAIL to VISIT_OCCURRENCE.

For example: an entire inpatient stay maybe one record in the VISIT_OCCURRENCE table. This may have one or more detail records such as ER, ICU, medical floor, rehabilitation floor etc. Each of these visit details may have different start/end date-times, different concept_ids and fact_ids. These would become separate records in VISIT_DETAIL with a FK link to VISIT_OCCURRENCE.

Each record within VISIT_DETAIL may be related to each other, sequentially â€“> ER leading to ICU leading to medical floor, leading to rehabilitation, or in hierarchical parent-child visit â€“> a visit for dialysis while in ICU.

Note the domain is Visit, and it is shared between VISIT_OCCURRENCE and VISIT_DETAIL in OMOP CDM. The key deviation from VISIT_OCCURRENCE is - self-referencing key: a new foreign key visit_detail_parent_id allows self referencing for nested visits. - VISIT_DETAIL points to its parent record in the VISIT_OCCURRENCE table (visit_occurrence_id)

6.5 CONDITION_OCCURRENCE

Conditions are records of a Person suggesting the presence of a disease or medical condition stated as a diagnosis, a sign, or a symptom, which is either observed by a Provider or reported by the patient. Conditions are recorded in different sources and levels of standardization, for example:

- Medical claims data include diagnoses coded in Source Vocabularies such as ICD-9-CM that are submitted as part of a reimbursement claim for health services
- EHRs may capture Person conditions in the form of diagnosis codes or symptoms

Field	Required	Type	Description
condition_occurrence_id	Yes	bigint	A unique identifier for each Condition Occurrence event.

Field	Required	Type	Description
person_id	Yes	bigint	A foreign key identifier to the Person who is experiencing the condition. The demographic details of that Person are stored in the PERSON table.
condition_concept_id	Yes	integer	A foreign key that refers to a Standard Concept identifier in the Standardized Vocabularies belonging to the 'Condition' domain.
condition_start_date	No	date	The date when the instance of the Condition is recorded.
condition_start_datetime	Yes	datetime	The date and time when the instance of the Condition is recorded.
condition_end_date	No	date	The date when the instance of the Condition is considered to have ended.
condition_end_datetime	No	datetime	The date when the instance of the Condition is considered to have ended.
condition_type_concept_id	Yes	integer	A foreign key to the predefined Concept identifier in the Standardized Vocabularies reflecting the source data from which the Condition was recorded, the level of standardization, and the type of occurrence. These belong to the 'Condition Type' vocabulary
condition_status_concept_id	Yes	integer	A foreign key that refers to a Standard Concept identifier in the Standardized Vocabularies reflecting the point of care at which the Condition was diagnosed.
stop_reason	No	varchar(20)	The reason that the Condition was no longer present, as indicated in the source data.
provider_id	No	integer	A foreign key to the Provider in the PROVIDER table who was responsible for capturing (diagnosing) the Condition.
visit_occurrence_id	No	integer	A foreign key to the visit in the VISIT_OCCURRENCE table during which the Condition was determined (diagnosed).
visit_detail_id	No	integer	A foreign key to the visit in the VISIT_DETAIL table during which the Condition was determined (diagnosed).
condition_source_value	No	varchar(50)	The source code for the Condition as it appears in the source data. This code is mapped to a Standard Condition Concept in the Standardized Vocabularies and the original code is stored here for reference.
condition_source_concept_id	Yes	integer	A foreign key to a Condition Concept that refers to the code used in the source.
condition_status_source_value	No	varchar(50)	The source code for the condition status as it appears in the source data. This code is mapped to a Standard Concept in the Standardized Vocabularies and the original code is stored here for reference.

6.5.1 Conventions

No.	Convention Description
1	Valid Condition Concepts belong to the 'Condition' domain.
2	Condition records are typically inferred from diagnostic codes recorded in the source data. Such code systems, like ICD-9-CM, ICD-10-CM, Read etc., provide a comprehensive coverage of conditions. However, if the diagnostic code in the source does not define a condition, but rather an observation or a procedure, then such information is not stored in the CONDITION_OCCURRENCE table, but in the respective tables indicated by the domain.
3	Source Condition identifiers are mapped to Standard Concepts for Conditions in the Standardized Vocabularies. When the source code cannot be translated into a Standard Concept, a CONDITION_OCCURRENCE entry is stored with only the corresponding SOURCE_CONCEPT_ID and SOURCE_VALUE, while the CONDITION_CONCEPT_ID is set to 0.
4	Family history and past diagnoses ('history of') are not recorded in the CONDITION_OCCURRENCE table. Instead, they are listed in the OBSERVATION table.
5	Codes written in the process of establishing the diagnosis, such as 'question of' of and 'rule out', are not represented here. Instead, they are listed in the OBSERVATION table, if they are used for analyses.

| 6 | A Condition Occurrence Type is assigned based on the data source and type of condition attribute, for example:

ICD-9-CM Primary Diagnosis from inpatient and outpatient claims

ICD-9-CM Secondary Diagnoses from inpatient and outpatient claims

Diagnoses or problems recorded in an EHR.

| | 7 | Valid Condition Occurrence Type Concepts belong to the 'Condition Type' vocabulary in the 'Type Concept' domain. | 8 | The Stop Reason indicates why a Condition is no longer valid with respect to the purpose within the source data. Typical values include 'Discharged', 'Resolved', etc. Note that a Stop Reason does not necessarily imply that the condition is no longer occurring. | 9 | Condition source codes are typically ICD-9-CM, Read or ICD-10-CM diagnosis codes from medical claims or discharge status/visit diagnosis codes from EHRs. | 10 | Presently, there is no designated vocabulary, domain, or class that represents condition status. The following concepts from SNOMED are recommended:

Admitting diagnosis: 4203942

Final diagnosis: 4230359 (should also be used for discharge diagnosis

Preliminary diagnosis: 4033240

|

6.6 DEATH

As of OMOP CDM v6.0, the DEATH table has been deprecated in favor of storing the cause of death in the CONDITION_OCCURRENCE table, any observations relating to death stored in the OBSERVATION table, and a singular death date will be chosen and stored in the PERSON table.

The 'Death' domain contains the clinical events surrounding how and when a Person dies. A Person can have information in the source system containing evidence about the Death, such as:

- Condition Code in the Header or Detail information of claims
- Status of enrollment into a health plan
- Explicit record in EHR data

6.6.1 Conventions

No.	Convention Description
1	Living patients should not have a value in PERSON.DEATH_DATETIME, nor should they have any records relating to death either in the CONDITION_OCCURRENCE or OBSERVATION tables
2	Only one death date per individual can be used. If a patient has clinical activity (e.g. prescriptions filled, labs performed, etc) more than 60+ days after death you may want to drop the death record as it may have been falsely reported. If multiple records of death exist on multiple days you may select the death that you deem most reliable (e.g. death at discharge) or select the latest death date (THEMIS issue #6).
3	If multiple death records occur, the date and the person have to be the same, but the cause can be different. Can be reported by different sources as well (THEMIS issue #5).
4	If PERSON.DEATH_DATETIME cannot be precisely determined from the data, the best approximation should be used.
5	Any cause of death should be stored in the CONDITION_OCCURRENCE table, using the CONDITION_TYPE vocabulary with the DEATH_TYPE concept class.
6	All observations relating to death should be stored in the OBSERVATION table, including the concept 4306655.

| 7 | The DEATH_DATETIME in the PERSON table should not be used as the way to find all deaths

`select * from PERSON where death_datetime is not null` should not be the practice

Rather, deaths should be found through the OBSERVATION table and the PERSON table is only used to determine which death date should be used in analysis

6.7 DRUG_EXPOSURE

The ‘Drug’ domain captures records about the utilization of a Drug when ingested or otherwise introduced into the body. A Drug is a biochemical substance formulated in such a way that when administered to a Person it will exert a certain physiological effect. Drugs include prescription and over-the-counter medicines, vaccines, and large-molecule biologic therapies. Radiological devices ingested or applied locally do not count as Drugs.

Drug Exposure is inferred from clinical events associated with orders, prescriptions written, pharmacy dispensings, procedural administrations, and other patient-reported information, for example:

- The ‘Prescription’ section of an EHR captures prescriptions written by physicians or from electronic ordering systems
- The ‘Medication list’ section of an EHR for both non-prescription products and medications prescribed by other providers
- Prescriptions filled at dispensing providers such as pharmacies, and then captured in reimbursement claim systems
- Drugs administered as part of a Procedure, such as chemotherapy or vaccines.

Field	Required	Type	Description
drug_exposure_id	Yes	bigint	A system-generated unique identifier for each Drug utilization event.
person_id	Yes	bigint	A foreign key identifier to the Person who is subjected to the Drug. The demographic details of that Person are stored in the PERSON table.
drug_concept_id	Yes	integer	A foreign key that refers to a Standard Concept identifier in the Standardized Vocabularies belonging to the ‘Drug’ domain.
drug_exposure_start_date	No	date	The start date for the current instance of Drug utilization. Valid entries include a start date of a prescription, the date a prescription was filled, or the date on which a Drug administration procedure was recorded.
drug_exposure_start_datetime	Yes	datetime	The start date and time for the current instance of Drug utilization. Valid entries include a start datetime of a prescription, the date and time a prescription was filled, or the date and time on which a Drug administration procedure was recorded.
drug_exposure_end_date	No	date	The end date for the current instance of Drug utilization. Depending on different sources, it could be a known or an inferred date and denotes the last day at which the patient was still exposed to Drug.
drug_exposure_end_datetime	No	datetime	The end date and time for the current instance of Drug utilization. Depending on different sources, it could be a known or an inferred date and time and denotes the last day at which the patient was still exposed to Drug.
verbatim_end_date	No	date	The known end date of a drug_exposure as provided by the source.

Field	Required	Type	Description
drug_type_concept_id	Yes	integer	A foreign key to the predefined Concept identifier in the Standardized Vocabularies reflecting the type of Drug Exposure recorded. It indicates how the Drug Exposure was represented in the source data and belongs to the 'Drug Type' vocabulary.
stop_reason	No	varchar(20)	The reason the Drug was stopped. Reasons include regimen completed, changed, removed, etc.
refills	No	integer	The number of refills after the initial prescription. The initial prescription is not counted, values start with null.
quantity	No	float	The quantity of drug as recorded in the original prescription or dispensing record.
days_supply	No	integer	The number of days of supply of the medication as prescribed. This reflects the intention of the provider for the length of exposure.
sig	No	varchar(MAX)	The directions ('signetur') on the Drug prescription as recorded in the original prescription (and printed on the container) or dispensing record.
route_concept_id	Yes	integer	A foreign key that refers to a Standard Concept identifier in the Standardized Vocabularies reflecting the route of administration and belonging to the 'Route' domain.
lot_number	No	varchar(50)	An identifier assigned to a particular quantity or lot of Drug product from the manufacturer.
provider_id	No	integer	A foreign key to the provider in the PROVIDER table who initiated (prescribed or administered) the Drug Exposure.
visit_occurrence_id	No	integer	A foreign key to the Visit in the VISIT_OCCURRENCE table during which the Drug Exposure was initiated.
visit_detail_id	No	integer	A foreign key to the Visit Detail in the VISIT_DETAIL table during which the Drug Exposure was initiated.
drug_source_value	No	varchar(50)	The source code for the Drug as it appears in the source data. This code is mapped to a Standard Drug concept in the Standardized Vocabularies and the original code is, stored here for reference.
drug_source_concept_id	Yes	integer	A foreign key to a Drug Concept that refers to the code used in the source.
route_source_value	No	varchar(50)	The information about the route of administration as detailed in the source.
dose_unit_source_value	No	varchar(50)	The information about the dose unit as detailed in the source.

6.7.1 Conventions

No.	Convention Description
1	Valid Concepts for the DRUG_CONCEPT_ID field belong to the 'Drug' domain. Most Concepts in the Drug domain are based on RxNorm, but some may come from other sources. Concepts are members of the Clinical Drug or Pack, Branded Drug or Pack, Drug Component or Ingredient classes.
2	Source drug identifiers, including NDC codes, Generic Product Identifiers, etc. are mapped to Standard Drug Concepts in the Standardized Vocabularies (e.g., based on RxNorm). When the Drug Source Value of the code cannot be translated into Standard Drug Concept IDs, a Drug exposure entry is stored with only the corresponding SOURCE_CONCEPT_ID and DRUG_SOURCE_VALUE and a DRUG_CONCEPT_ID of 0.
3	The Drug Concept with the most detailed content of information is preferred during the mapping process. These are indicated in the CONCEPT_CLASS_ID field of the Concept and are recorded in the following order of precedence: 'Branded Pack', 'Clinical Pack', 'Branded Drug', 'Clinical Drug', 'Branded Drug Component', 'Clinical Drug Component', 'Branded Drug Form', 'Clinical Drug Form', and only if no other information is available 'Ingredient'. Note: If only the drug class is known, the DRUG_CONCEPT_ID field should contain 0.
4	A Drug Type is assigned to each Drug Exposure to track from what source the information was drawn or inferred from. The valid CONCEPT_CLASS_ID for these Concepts is 'Drug Type'.
5	The content of the refills field determines the current number of refills, not the number of remaining refills. For example, for a drug prescription with 2 refills, the content of this field for the 3 Drug Exposure events are null, 1 and 2.
6	The ROUTE_CONCEPT_ID refers to a Standard Concepts of the 'Route' domain. Note: Route information can also be inferred from the Drug product itself by determining the Drug Form of the Concept, creating some partial overlap of the same type of information. Therefore, route information should be stored in DRUG_CONCEPT_ID (as a drug with corresponding Dose Form). The ROUTE_CONCEPT_ID could be used for storing more granular forms e.g. 'Intraventricular cardiac'.
7	The LOT_NUMBER field contains an identifier assigned from the manufacturer of the Drug product.

No.	Convention Description
8	If possible, the visit in which the drug was prescribed or delivered is recorded in the VISIT_OCCURRENCE_ID field through a reference to the visit table.
9	If possible, the prescribing or administering provider (physician or nurse) is recorded in the PROVIDER_ID field through a reference to the provider table.
10	The DRUG_EXPOSURE_END_DATE denotes the day the drug exposure ended for the patient. This could be that the duration of DRUG_SUPPLY was reached (in which case DRUG_EXPOSURE_END_DATETIME = DRUG_EXPOSURE_START_DATETIME + DAYS_SUPPLY -1 day), or because the exposure was stopped (medication changed, medication discontinued, etc.)
11	When the native data suggests a drug exposure has a days supply less than 0, drop the record as unknown if a person has received the drug or not (THEMIS issue #24).
12	If a patient has multiple records on the same day for the same drug or procedures the ETL should not de-dupe them unless there is probable reason to believe the item is a true data duplicate (THEMIS issue #14).

6.8 PROCEDURE_OCCURRENCE

The PROCEDURE_OCCURRENCE table contains records of activities or processes ordered by, or carried out by, a healthcare provider on the patient to have a diagnostic or therapeutic purpose. Procedures are present in various data sources in different forms with varying levels of standardization. For example:

- Medical Claims include procedure codes that are submitted as part of a claim for health services rendered, including procedures performed.
- Electronic Health Records that capture procedures as orders.

Field	Required	Type	Description
procedure_occurrence_id	Yes	integer	A system-generated unique identifier for each Procedure Occurrence.
person_id	Yes	integer	A foreign key identifier to the Person who is subjected to the Procedure. The demographic details of that Person are stored in the PERSON table.
procedure_concept_id	Yes	integer	A foreign key that refers to a standard procedure Concept identifier in the Standardized Vocabularies.
procedure_date	No	date	The date on which the Procedure was performed.
procedure_datetime	Yes	datetime	The date and time on which the Procedure was performed.

Field	Required	Type	Description
procedure_type_concept_id	Yes	integer	A foreign key to the predefined Concept identifier in the Standardized Vocabularies reflecting the type of source data from which the procedure record is derived, belonging to the 'Procedure Type' vocabulary.
modifier_concept_id	Yes	integer	A foreign key to a Standard Concept identifier for a modifier to the Procedure (e.g. bilateral). These concepts are typically distinguished by 'Modifier' concept classes (e.g., 'CPT4 Modifier' as part of the 'CPT4' vocabulary).
quantity	No	integer	The quantity of procedures ordered or administered.
provider_id	No	integer	A foreign key to the provider in the PROVIDER table who was responsible for carrying out the procedure.
visit_occurrence_id	No	integer	A foreign key to the Visit in the VISIT_OCCURRENCE table during which the Procedure was carried out.
visit_detail_id	No	integer	A foreign key to the Visit Detail in the VISIT_DETAIL table during which the Procedure was carried out.
procedure_source_value	No	varchar(50)	The source code for the Procedure as it appears in the source data. This code is mapped to a standard procedure Concept in the Standardized Vocabularies and the original code is, stored here for reference. Procedure source codes are typically ICD-9-Proc, CPT-4, HCPCS or OPCS-4 codes.
procedure_source_concept_id	Yes	integer	A foreign key to a Procedure Concept that refers to the code used in the source.
modifier_source_value	No	varchar(50)	The source code for the qualifier as it appears in the source data.

6.8.1 Conventions

No.	Convention Description
1	Valid Procedure Concepts belong to the 'Procedure' domain. Procedure Concepts are based on a variety of vocabularies: SNOMED-CT, ICD-9-Proc, CPT-4, HCPCS and OPCS-4, but also atypical Vocabularies such as ICD-9-CM or MedDRA.
2	Procedures are expected to be carried out within one day and therefore have no end date.

No.	Convention Description
3	Procedures could involve the application of a drug, in which case the procedural component is recorded in the procedure table and simultaneously the administered drug in the drug exposure table when both the procedural component and drug are identifiable.
4	If the quantity value is omitted, a single procedure is assumed.
5	The Procedure Type defines from where the Procedure Occurrence is drawn or inferred. For administrative claims records the type indicates whether a Procedure was primary or secondary and their relative positioning within a claim.
6	The Visit during which the procedure was performed is recorded through a reference to the VISIT_OCCURRENCE table. This information is not always available.
7	The Visit Detail during with the procedure was performed is recorded through a reference to the VISIT_DETAIL table. This information is not always available.
8	The Provider carrying out the procedure is recorded through a reference to the PROVIDER table. This information is not always available.

| 9 | When dealing with duplicate records, the ETL must determine whether to sum them up into one record or keep them separate. Things to consider are:

Same Procedure

Same PROCEDURE_DATETIME

Same Visit Occurrence or Visit Detail

Same Provider

Same Modifier for Procedures

Same COST_ID

THEMIS issue #27 | | 10 | If a Procedure has a quantity of ‘0’ in the source, this should default to ‘1’ in the ETL. If there is a record in the source it can be assumed the exposure occurred at least once (THEMIS issue #26).|

6.9 DEVICE_EXPOSURE

The ‘Device’ domain captures information about a person’s exposure to a foreign physical object or instrument which is used for diagnostic or therapeutic purposes through a mechanism beyond chemical action. Devices include implantable objects (e.g. pacemakers, stents, artificial joints), medical equipment and supplies (e.g. bandages, crutches, syringes), other instruments used in medical procedures (e.g. sutures, defibrillators) and material used in clinical care (e.g. adhesives, body material, dental material, surgical material).

Field	Required	Type	Description
device_exposure_id	Yes	bigint	A system-generated unique identifier for each Device Exposure.
person_id	Yes	bigint	A foreign key identifier to the Person who is subjected to the Device. The demographic details of that Person are stored in the PERSON table.
device_concept_id	Yes	integer	A foreign key that refers to a Standard Concept identifier in the Standardized Vocabularies belonging to the 'Device' domain.
device_exposure_start_date	No	date	The date the Device or supply was applied or used.
device_exposure_start_datetime	Yes	datetime	The date and time the Device or supply was applied or used.
device_exposure_end_date	No	date	The date use of the Device or supply was ceased.
device_exposure_end_datetime	No	datetime	The date and time use of the Device or supply was ceased.
device_type_concept_id	Yes	integer	A foreign key to the predefined Concept identifier in the Standardized Vocabularies reflecting the type of Device Exposure recorded. It indicates how the Device Exposure was represented in the source data and belongs to the 'Device Type' domain.
unique_device_id	No	varchar(50)	A UDI or equivalent identifying the instance of the Device used in the Person.
quantity	No	integer	The number of individual Devices used in the exposure.
provider_id	No	integer	A foreign key to the provider in the PROVIDER table who initiated or administered the Device.
visit_occurrence_id	No	integer	A foreign key to the visit in the VISIT_OCCURRENCE table during which the Device was used.
visit_detail_id	No	integer	A foreign key to the visit detail record in the VISIT_DETAIL table during which the Device was used.
device_source_value	No	varchar(50)	The source code for the Device as it appears in the source data. This code is mapped to a Standard Device Concept in the Standardized Vocabularies and the original code is stored here for reference.
device_source_concept_id	Yes	integer	A foreign key to a Device Concept that refers to the code used in the source.

6.9.1 Conventions

No.	Convention Description
1	The distinction between Devices or supplies and Procedures are sometimes blurry, but the former are physical objects while the latter are actions, often to apply a Device or supply.
2	For medical devices that are regulated by the FDA, a Unique Device Identification (UDI) is provided if available in the data source and is recorded in the UNIQUE_DEVICE_ID field.
3	Valid Device Concepts belong to the 'Device' domain. The Concepts of this domain are derived from the DI portion of a UDI or based on other source vocabularies, like HCPCS.
4	A Device Type is assigned to each Device Exposure to track from what source the information was drawn or inferred. The valid vocabulary for these Concepts is 'Device Type'.
5	The Visit during which the Device was first used is recorded through a reference to the VISIT_OCCURRENCE table.
6	The Visit Detail during which the Device was first used is recorded through a reference to the VISIT_DETAIL table.
7	The Provider exposing the patient to the Device is recorded through a reference to the PROVIDER table.

| 8 | When dealing with duplicate records, the ETL must determine whether to sum them up into one record or keep them separate. Things to consider are:

Same Device/Procedure

Same DEVICE_EXPOSURE_START_DATETIME

Same Visit Occurrence or Visit Detail

Same Provider

Same Modifier for Procedures

Same COST_ID

THEMIS issue #27 | | 9 | If a Device Exposure has a quantity of '0' in the source, this should default to '1' in the ETL. If there is a record in the source it can be assumed the exposure occurred at least once (THEMIS issue #26). |

6.10 MEASUREMENT

The MEASUREMENT table contains records of Measurement, i.e. structured values (numerical or categorical) obtained through systematic and standardized examination or testing of a Person or Person's sample. The MEASUREMENT table contains both orders and results of such Measurements as laboratory tests, vital signs, quantitative findings from pathology reports, etc.

Field	Required	Type	Description
measurement_id	Yes	integer	A unique identifier for each Measurement.

Field	Required	Type	Description
person_id	Yes	integer	A foreign key identifier to the Person about whom the measurement was recorded. The demographic details of that Person are stored in the PERSON table.
measurement_concept_id	Yes	integer	A foreign key to the standard measurement concept identifier in the Standardized Vocabularies. These belong to the 'Measurement' domain, but could overlap with the 'Observation' domain (see #3 below).
measurement_date	No	date	The date of the Measurement.
measurement_datetime	Yes	datetime	The date and time of the Measurement. Some database systems don't have a datatype of time. To accommodate all temporal analyses, datatype datetime can be used (combining measurement_date and measurement_time forum discussion)
measurement_time	No	varchar(10)	The time of the Measurement. This is present for backwards compatibility and will be deprecated in an upcoming version
measurement_type_concept_id	Yes	integer	A foreign key to the predefined Concept in the Standardized Vocabularies reflecting the provenance from where the Measurement record was recorded. These belong to the 'Meas Type' vocabulary
operator_concept_id	No	integer	A foreign key identifier to the predefined Concept in the Standardized Vocabularies reflecting the mathematical operator that is applied to the value_as_number. Operators are <, <=, =, >=, > and these concepts belong to the 'Meas Value Operator' domain.
value_as_number	No	float	A Measurement result where the result is expressed as a numeric value.
value_as_concept_id	No	integer	A foreign key to a Measurement result represented as a Concept from the Standardized Vocabularies (e.g., positive/negative, present/absent, low/high, etc.). These belong to the 'Meas Value' domain
unit_concept_id	No	integer	A foreign key to a Standard Concept ID of Measurement Units in the Standardized Vocabularies that belong to the 'Unit' domain.
range_low	No	float	The lower limit of the normal range of the Measurement result. The lower range is assumed to be of the same unit of measure as the Measurement value.
range_high	No	float	The upper limit of the normal range of the Measurement. The upper range is assumed to be of the same unit of measure as the Measurement value.

Field	Required	Type	Description
provider_id	No	integer	A foreign key to the provider in the PROVIDER table who was responsible for initiating or obtaining the measurement.
visit_occurrence_id	No	integer	A foreign key to the Visit in the VISIT_OCCURRENCE table during which the Measurement was recorded.
visit_detail_id	No	integer	A foreign key to the Visit Detail in the VISIT_DETAIL table during which the Measurement was recorded.
measurement_source_value	No	varchar(50)	The Measurement name as it appears in the source data. This code is mapped to a Standard Concept in the Standardized Vocabularies and the original code is stored here for reference.
measurement_source_concept_id	Yes	integer	A foreign key to a Concept in the Standard Vocabularies that refers to the code used in the source.
unit_source_value	No	varchar(50)	The source code for the unit as it appears in the source data. This code is mapped to a standard unit concept in the Standardized Vocabularies and the original code is stored here for reference.
value_source_value	No	varchar(50)	The source value associated with the content of the value_as_number or value_as_concept_id as stored in the source data.

6.10.1 Conventions

No.	Convention Description
1	Measurements differ from Observations in that they require a standardized test or some other activity to generate a quantitative or qualitative result. For example, LOINC 1755-8 concept_id 3027035 'Albumin [Mass/time] in 24 hour Urine' is the lab test to measure a certain chemical in a urine sample.
2	Even though each Measurement always have a result, the fields VALUE_AS_NUMBER and VALUE_AS_CONCEPT_ID are not mandatory. When the result is not known, the Measurement record represents just the fact that the corresponding Measurement was carried out, which in itself is already useful information for some use cases.

No.	Convention Description
3	Valid Measurement Concepts (MEASUREMENT_CONCEPT_ID) belong to the ‘Measurement’ domain, but could overlap with the ‘Observation’ domain. This is due to the fact that there is a continuum between systematic examination or testing (Measurement) and a simple determination of fact (Observation). When the Measurement Source Value of the code cannot be translated into a standard Measurement Concept ID, a Measurement entry is stored with only the corresponding SOURCE_CONCEPT_ID and MEASUREMENT_SOURCE_VALUE and a MEASUREMENT_CONCEPT_ID of 0.
4	Measurements are stored as attribute value pairs, with the attribute as the Measurement Concept and the value representing the result. The value can be a Concept (stored in VALUE_AS_CONCEPT), or a numerical value (VALUE_AS_NUMBER) with a Unit (UNIT_CONCEPT_ID).
5	Valid Concepts for the VALUE_AS_CONCEPT field belong to the ‘Meas Value’ domain.
6	For some Measurement Concepts, the result is included in the test. For example, ICD10 concept_id 45595451 ‘Presence of alcohol in blood, level not specified’ indicates a Measurement and the result (present). In those situations, the CONCEPT_RELATIONSHIP table in addition to the ‘Maps to’ record contains a second record with the relationship_id set to ‘Maps to value’. In this example, the ‘Maps to’ relationship directs to 4041715 ‘Blood ethanol measurement’ as well as a ‘Maps to value’ record to 4181412 ‘Present’.
7	The OPERATOR_CONCEPT_ID is optionally given for relative Measurements where the precise value is not available but its relation to a certain benchmarking value is. For example, this can be used for minimal detection thresholds of a test.
8	The meaning of Concept 4172703 for ‘=’ is identical to omission of a OPERATOR_CONCEPT_ID value. Since the use of this field is rare, it’s important when devising analyses to not to forget testing for the content of this field for values different from =.
9	Valid Concepts for the OPERATOR_CONCEPT_ID field belong to the ‘Meas Value Operator’ domain.
10	The Unit is optional even if a VALUE_AS_NUMBER is provided.
11	If reference ranges for upper and lower limit of normal as provided (typically by a laboratory) these are stored in the RANGE_HIGH and RANGE_LOW fields. Ranges have the same unit as the VALUE_AS_NUMBER.

No.	Convention Description
12	The Visit during which the observation was made is recorded through a reference to the VISIT_OCCURRENCE table. This information is not always available.
13	The Provider making the observation is recorded through a reference to the PROVIDER table. This information is not always available.

| 14 | If there is a negative value coming from the source, set the VALUE_AS_NUMBER to NULL, with the exception of the following Measurements (listed as LOINC codes):

1925-7 Base excess in Arterial blood by calculation

1927-3 Base excess in Venous blood by calculation

8632-2 QRS-Axis

11555-0 Base excess in Blood by calculation

1926-5 Base excess in Capillary blood by calculation

28638-5 Base excess in Arterial cord blood by calculation

28639-3 Base excess in Venous cord blood by calculation

THEMIS issue #16 |

6.11 NOTE

The NOTE table captures unstructured information that was recorded by a provider about a patient in free text notes on a given date.

Field	Required	Type	Description
note_id	Yes	integer	A unique identifier for each note.
person_id	Yes	integer	A foreign key identifier to the Person about whom the Note was recorded. The demographic details of that Person are stored in the PERSON table.
note_event_id	No	integer	A foreign key identifier to the event (e.g. Measurement, Procedure, Visit, Drug Exposure, etc) record during which the note was recorded.
note_event_field_concept_id	No	integer	A foreign key to the predefined Concept in the Standardized Vocabularies reflecting the field to which the note_event_id is referring.
note_date	No	date	The date the note was recorded.
note_datetime	Yes	datetime	The date and time the note was recorded.
note_type_concept_id	Yes	integer	A foreign key to the predefined Concept in the Standardized Vocabularies reflecting the type, origin or provenance of the Note. These belong to the 'Note Type' vocabulary
note_class_concept_id	Yes	integer	A foreign key to the predefined Concept in the Standardized Vocabularies reflecting the HL7 LOINC Document Type Vocabulary classification of the note.
note_title	No	varchar(250)	The title of the Note as it appears in the source.

Field	Required	Type	Description
note_text	Yes	varchar(MAX)	The content of the Note.
encoding_concept_id	Yes	integer	A foreign key to the predefined Concept in the Standardized Vocabularies reflecting the note character encoding type
language_concept_id	Yes	integer	A foreign key to the predefined Concept in the Standardized Vocabularies reflecting the language of the note
provider_id	No	integer	A foreign key to the Provider in the PROVIDER table who took the Note.
visit_occurrence_id	No	integer	A foreign key to the Visit in the VISIT_OCCURRENCE table when the Note was taken.
visit_detail_id	No	integer	A foreign key to the Visit in the VISIT_DETAIL table when the Note was taken.
note_source_value	No	varchar(50)	The source value associated with the origin of the Note

6.11.1 Conventions

No.	Convention Description
1	The NOTE table contains free text (in ASCII, or preferably in UTF8 format) taken by a healthcare Provider.
2	The Visit during which the note was written is recorded through a reference to the VISIT_OCCURRENCE table. This information is not always available.
3	The Provider making the note is recorded through a reference to the PROVIDER table. This information is not always available.
4	The type of note_text is CLOB or varchar(MAX) depending on RDBMS
5	NOTE_CLASS_CONCEPT_ID is a foreign key to the CONCEPT table to describe a standardized combination of five LOINC axes (role, domain, setting, type of service, and document kind). See below for description.

6.11.2 Mapping of clinical documents to Clinical Document Ontology (CDO) and standard terminology

HL7/LOINC CDO is a standard for consistent naming of documents to support a range of use cases: retrieval, organization, display, and exchange. It guides the creation of LOINC codes for clinical notes. CDO annotates each document with 5 dimensions:

- **Kind of Document:** Characterizes the general structure of the document at a macro level (e.g. Anesthesia Consent)
- **Type of Service**: Characterizes the kind of service or activity (e.g. evaluations, consultations, and summaries). The notion of time sequence, e.g., at the beginning (admission) at the end (discharge) is subsumed in this axis. Example: Discharge Teaching.

- **Setting:** Setting is an extension of CMSï¿½s definitions (e.g. Inpatient, Outpatient)
- **Subject Matter Domain (SMD):** Characterizes the subject matter domain of a note (e.g. Anesthesiology)
- **Role:** Characterizes the training or professional level of the author of the document, but does not break down to specialty or subspecialty (e.g. Physician)

Each combination of these 5 dimensions rolls up to a unique LOINC code.

- According to CDO requirements, only 2 of the 5 dimensions are required to properly annotate a document: Kind of Document and any one of the other 4 dimensions.
- However, not all the permutations of the CDO dimensions will necessarily yield an existing LOINC code.2 HL7/LOINC workforce is committed to establish new LOINC codes for each new encountered combination of CDO dimensions.
- The full document ontology as it exists in the Vocabulary is too extensive to list here, but it is possible to explore through the ATHENA tool starting with the 'LOINC Document Ontology - Type of Service and Kind of Document' by walking through the 'Is a'/'Subsumes' relationship hierarchies.

6.12 NOTE_NLP

The NOTE_NLP table will encode all output of NLP on clinical notes. Each row represents a single extracted term from a note.

Field	Required	Type	Description
note_nlp_id	Yes	integer	A unique identifier for each term extracted from a note.
note_id	Yes	integer	A foreign key to the Note table note the term was
section_concept_id	Yes	integer	A foreign key to the predefined Concept in the Standardized Vocabularies representing the section of the extracted term.
snippet	No	varchar(250)	A small window of text surrounding the term.
offset	No	varchar(50)	Character offset of the extracted term in the input note.
lexical_variant	Yes	varchar(250)	Raw text extracted from the NLP tool.
note_nlp_concept_id	Yes	integer	A foreign key to the predefined Concept in the Standardized Vocabularies reflecting the normalized concept for the extracted term. Domain of the term is represented as part of the Concept table.
note_nlp_source_concept_id	Yes	integer	A foreign key to a Concept that refers to the code in the source vocabulary used by the NLP system
nlp_system	No	varchar(250)	Name and version of the NLP system that extracted the term.Useful for data provenance.
nlp_date	Yes	date	The date of the note processing.Useful for data provenance.
nlp_datetime	No	datetime	The date and time of the note processing. Useful for data provenance.
term_exists	No	varchar(1)	A summary modifier that signifies presence or absence of the term for a given patient. Useful for quick querying.

Field	Required	Type	Description
term_temporal	No	varchar(50)	An optional time modifier associated with the extracted term. (for now â€œpastâ€ or â€œpresentâ€ only). Standardize it later.
term_modifiers	No	varchar(2000)	A compact description of all the modifiers of the specific term extracted by the NLP system. (e.g. â€œson has rashâ€ ? â€œnegated=no,subject=family, certainty=undef,conditional=false,general=falseâ€).

6.12.1 Conventions

No.	Convention Description

| 1 | Term_exists is defined as a flag that indicates if the patient actually has or had the condition. Any of the following modifiers would make Term_exists false:

Negation = true

Subject = [anything other than the patient]

Conditional = true/li>

Rule_out = true

Uncertain = very low certainty or any lower certainties

A complete lack of modifiers would make Term_exists true.

For the modifiers that are there, they would have to have these values:

Negation = false

Subject = patient

Conditional = false

Rule_out = false

Uncertain = true or high or moderate or even low (could argue about low)

| | 2 | Term_temporal is to indicate if a condition is â€œpresentâ€ or just in the â€œpastâ€ . The following would be past:

History = true

Concept_date = anything before the time of the report

3 | Term_modifiers will concatenate all modifiers for different types of entities (conditions, drugs, labs etc) into one string. Lab values will be saved as one of the modifiers. A list of allowable modifiers (e.g., signature for medications) and their possible values will be standardized later. |

6.13 SURVEY_CONDUCT

7 SURVEY_CONDUCT

The SURVEY_CONDUCT table is used to store an instance of a completed survey or questionnaire. It captures details of the individual questionnaire such as who completed it, when it was completed and to which patient treatment or visit it relates to (if any). Each SURVEY has a SURVEY_CONCEPT_ID, a concept in the CONCEPT table identifying the questionnaire e.g. EQ5D, VR12, SF12. Each questionnaire should exist in the CONCEPT table. Each SURVEY can be optionally related to a specific patient visit in order to link it both to the visit during which it was completed and any subsequent visit where treatment was assigned based on the patient's responses.

Field	Required	Type	Description
SURVEY_CONDUCT_ID	Yes	integer	Unique identifier for each completed survey.
PERSON_ID	Yes	integer	A foreign key identifier to the Person in the PERSON table about whom the survey was completed.
SURVEY_CONCEPT_ID	Yes	integer	A foreign key to the predefined Concept identifier in the Standardized Vocabularies reflecting the name and identity of the survey.
SURVEY_START_DATE	No	date	Date on which the survey was started.
SURVEY_START_DATETIME	No	datetime	Date and time the survey was started.
SURVEY_END_DATE	Yes	date	Date on which the survey was completed.
SURVEY_END_DATETIME	No	datetime	Date and time the survey was completed.
PROVIDER_ID	NoÂ	integerÂ	A foreign key to the provider in the provider table who was associated with the survey completion.
ASSISTED_CONCEPT_ID	Yes	integer	A foreign key to the predefined Concept identifier in the Standardized Vocabularies indicating whether the survey was completed with assistance.
RESPONDENT_TYPE_CONCEPT_ID	Yes	integer	A foreign key to the predefined Concept identifier in the Standardized Vocabularies reflecting the respondent type. Example: Research Associate, Patient.
TIMING_CONCEPT_ID	Yes	integer	A foreign key to the predefined Concept identifier in the Standardized Vocabularies that refers to a certain timing. Example: 3 month follow-up, 6 month follow-up.
COLLECTION_METHOD_CONCEPT_ID	Yes	integer	A foreign key to the predefined Concept identifier in the Standardized Vocabularies reflecting the data collection method (e.g. Paper, Telephone, Electronic Questionnaire).

Field	Required	Type	Description
ASSISTED_SOURCE_VALUE	No	varchar(50)	Source value representing whether patient required assistance to complete the survey. Example: â€œCompleted without assistanceâ€ , â€ Completed with assistanceâ€ .
RESPONDENT_TYPE_SOURCE_VALUE	No	varchar(100)	Source code representing role of person who completed the survey.
TIMING_SOURCE_VALUE	No	varchar(100)	Text string representing the timing of the survey. Example: Baseline, 6-month follow-up.
COLLECTION_METHOD_SOURCE_VALUE	No	varchar(100)	The collection method as it appears in the source data.
SURVEY_SOURCE_VALUE	No	varchar(100)	The survey name/title as it appears in the source data.
SURVEY_SOURCE_CONCEPT_ID	Yes	integer	A foreign key to a predefined Concept that refers to the code for the survey name/title used in the source.
SURVEY_SOURCE_IDENTIFIER	No	varchar(100)	Unique identifier for each completed survey in source system.
VALIDATED_SURVEY_CONCEPT_ID	Yes	integer	A foreign key to the predefined Concept identifier in the Standardized Vocabularies reflecting the validation status of the survey.
VALIDATED_SURVEY_SOURCE_VALUE	No	integer	Source value representing the validation status of the survey.
SURVEY_VERSION_NUMBER	No	varchar(20)	Version number of the questionnaire or survey used.
VISIT_OCCURRENCE_ID	No	integer	A foreign key to the VISIT_OCCURRENCE table during which theÂ survey was completed
RESPONSE_VISIT_OCCURRENCE_ID	No	integerÂ	A foreign key to the visit in the VISIT_OCCURRENCE table during which treatment was carried out that relates to this survey.

7.0.1 Conventions

No.	Convention Description
1	Patient responses to survey questions are stored in the OBSERVATION table. Each record in the OBSERVATION table represents a single question/response pair and is linked to a specific SURVEY/questionnaire using OBSERVATION.DOMAIN_OCCURRENCE_ID and SURVEY.SURVEY_OCCURRENCE_ID.
2	Each response record is the response to a specific question identified by the OBSERVATION_CONCEPT_ID. This concept ID is a unique question contained in the CONCEPT table.

| 3 | An individual survey question can have multiple responses to a question (e.g. which of these items relate to you, a, b, c ,â€¦?). Each response is stored as a separate record in the OBSERVATION table.

The name (question) is stored as OBSERVATION_CONCEPT_ID and the value (answer) is stored as OBSERVATION_AS_CONCEPT_ID where the answer is categorical and is defined as a concept in the concept table, OBSERVATION_AS_NUMBER where the answer is numeric, OBSERVATION_AS_STRING where the answer is a free text string or OBSERVATION_AS_DATETIME. | 4 | The question / answer observation record is linked to the patient questionnaire used for collecting the data using two new fields in the OBSERVATION table; DOMAIN_ID and DOMAIN_OCCURRENCE_ID.

DOMAIN_ID for any survey related observations contains the text â€˜Surveyâ€™ and DOMAIN_OCCURRENCE_ID contains the SURVEY_OCCURRENCE_ID of the specific survey.

This domain construct can be used for other observation groupings.

| | 5 | The OBSERVATION table can also store survey scoring results. Many validated PRO questionnaires have scoring algorithms (many of which proprietary) that return an overall patient score based on the answers provided.

Survey scores are identified by their OBSERVATION_CONCEPT_ID and are linked back to the scored survey using the same DOMAIN construct described.

7.1 OBSERVATION

The OBSERVATION table captures clinical facts about a Person obtained in the context of examination, questioning or a procedure. Any data that cannot be represented by any other domains, such as social and lifestyle facts, medical history, family history, etc. are recorded here.

Field	Required	Type	Description
observation_id	Yes	integer	A unique identifier for each observation.
person_id	Yes	integer	A foreign key identifier to the Person about whom the observation was recorded. The demographic details of that Person are stored in the PERSON table.
observation_concept_id	Yes	integer	A foreign key to the standard observation concept identifier in the Standardized Vocabularies.
observation_date	No	date	The date of the observation.
observation_datetime	Yes	datetime	The date and time of the observation.
observation_type_concept_id	Yes	integer	A foreign key to the predefined concept identifier in the Standardized Vocabularies reflecting the type of the observation.
value_as_number	No	float	The observation result stored as a number. This is applicable to observations where the result is expressed as a numeric value.
value_as_string	No	varchar(60)	The observation result stored as a string. This is applicable to observations where the result is expressed as verbatim text.

Field	Required	Type	Description
value_as_concept_id	No	Integer	A foreign key to an observation result stored as a Concept ID. This is applicable to observations where the result can be expressed as a Standard Concept from the Standardized Vocabularies (e.g., positive/negative, present/absent, low/high, etc.).
qualifier_concept_id	No	integer	A foreign key to a Standard Concept ID for a qualifier (e.g., severity of drug-drug interaction alert)
unit_concept_id	No	integer	A foreign key to a Standard Concept ID of measurement units in the Standardized Vocabularies.
provider_id	No	integer	A foreign key to the provider in the PROVIDER table who was responsible for making the observation.
visit_occurrence_id	No	integer	A foreign key to the visit in the VISIT_OCCURRENCE table during which the observation was recorded.
visit_detail_id	No	integer	A foreign key to the visit in the VISIT_DETAIL table during which the observation was recorded.
observation_source_value	No	varchar(50)	The observation code as it appears in the source data. This code is mapped to a Standard Concept in the Standardized Vocabularies and the original code is, stored here for reference.
observation_source_concept_id	Yes	integer	A foreign key to a Concept that refers to the code used in the source.
unit_source_value	No	varchar(50)	The source code for the unit as it appears in the source data. This code is mapped to a standard unit concept in the Standardized Vocabularies and the original code is, stored here for reference.
qualifier_source_value	No	varchar(50)	The source value associated with a qualifier to characterize the observation
observation_event_id	No	integer	A foreign key to an event table (e.g., PROCEDURE_OCCURRENCE_ID).
obs_event_field_concept_id	Yes	integer	A foreign key that refers to a Standard Concept identifier in the Standardized Vocabularies referring to the field represented in the OBSERVATION_EVENT_ID.

Field	Required	Type	Description
value_as_datetime	No	integer	The observation result stored as a datetime value. This is applicable to observations where the result is expressed as a point in time.

7.1.1 Conventions

No.	Convention Description
1	Observations differ from Measurements in that they do not require a standardized test or some other activity to generate clinical fact. Typical observations are medical history, family history, the stated need for certain treatment, social circumstances, lifestyle choices, healthcare utilization patterns, etc. If the generation clinical facts requires a standardized testing such as lab testing or imaging and leads to a standardized result, the data item is recorded in the MEASUREMENT table. If the clinical fact observed determines a sign, symptom, diagnosis of a disease or other medical condition, it is recorded in the CONDITION_OCCURRENCE table.
2	Valid Observation Concepts are not enforced to be from any domain. They still should be Standard Concepts, and they typically belong to the 'Observation' or sometimes 'Measurement' domain.
3	Observations can be stored as attribute value pairs, with the attribute as the Observation Concept and the value representing the clinical fact. This fact can be a Concept (stored in VALUE_AS_CONCEPT), a numerical value (VALUE_AS_NUMBER), a verbatim string (VALUE_AS_STRING), or a datetime (VALUE_AS_DATETIME). Even though Observations do not have an explicit result, the clinical fact can be stated separately from the type of Observation in the VALUE_AS_* fields.
4	It is recommended for Observations that are suggestive statements of positive assertion should have a value of 'Yes' (concept_id=4188539), recorded, even though the null value is the equivalent.
5	Valid Concepts of the VALUE_AS_CONCEPT field are not enforced, but typically belong to the 'Meas Value' domain.
6	For numerical facts a Unit can be provided in the UNIT_CONCEPT_ID.
7	For facts represented as Concepts no domain membership is enforced.

No.	Convention Description
8	Note that the value of VALUE_AS_CONCEPT_ID may be provided through mapping from a source Concept which contains the content of the Observation. In those situations, the CONCEPT_RELATIONSHIP table in addition to the 'Maps to' record contains a second record with the relationship_id set to 'Maps to value'. For example, ICD9CM V17.5 concept_id 44828510 'Family history of asthma' has a 'Maps to' relationship to 4167217 'Family history of clinical finding' as well as a 'Maps to value' record to 317009 'Asthma'.
9	The QUALIFIER_CONCEPT_ID field contains all attributes specifying the clinical fact further, such as as degrees, severities, drug-drug interaction alerts etc.
10	The Visit during which the Observation was made is recorded through a reference to the VISIT_OCCURRENCE table. This information is not always available.
11	The Visit Detail during which the Observation was made is recorded through a reference to the VISIT_DETAIL table. This information is not always available.
12	The Provider making the observation is recorded through a reference to the PROVIDER table. This information is not always available.
13	When storing patient responses to survey questions, each record in the OBSERVATION table represents a single question/response pair and is linked to a specific survey/questionnaire using OBSERVATION.OBSERVATION_EVENT_ID and SURVEY_CONDUCT.SURVEY_CONDUCT_ID.
14	Each survey response record is the response to a specific question identified by the OBSERVATION_CONCEPT_ID. This concept ID is a unique question contained in the CONCEPT table.
15	An individual survey question can have multiple responses to a question (e.g. which of these items relate to you, a,b,c,...?). Each response is stored as a separate record in the OBSERVATION table.

| 16 | The question / answer OBSERVATION record is linked to the patient questionnaire used for collecting the data using two new fields in the OBSERVATION table; OBS_EVENT_FIELD_CONCEPT_ID and OBSERVATION_EVENT_ID.

OBS_EVENT_FIELD_CONCEPT_ID for any survey related observations contains the concept that refers to the field SURVEY_CONDUCT_ID and OBSERVATION_EVENT_ID contains the actual SURVEY_CONDUCT_ID of the specific survey

This construct can be used for other observation groupings

| | 17 | The OBSERVATION table can also store survey scoring results. Many validated PRO questionnaires have scoring algorithms (many of which proprietary) that return an overall patient score based on the answers

provided.

Survey scores are identified by their OBSERVATION_CONCEPT_ID and are linked back to the scored survey using the same EVENT_FIELD construct described above. In the name/value pair model, the name (question) is stored as OBSERVATION_CONCEPT_ID and the value (answer) is stored as OBSERVATION_AS_CONCEPT_ID where the answer is categorical and is defined as a concept in the concept table, OBSERVATION_AS_NUMBER where the answer is numeric, OBSERVATION_AS_STRING where the answer is a free text string or OBSERVATION_AS_DATETIME.

7.2 SPECIMEN

The specimen domain contains the records identifying biological samples from a person.

Field	Required	Type	Description
specimen_id	Yes	integer	A unique identifier for each specimen.
person_id	Yes	integer	A foreign key identifier to the Person for whom the Specimen is recorded.
specimen_concept_id	Yes	integer	A foreign key referring to a Standard Concept identifier in the Standardized Vocabularies for the Specimen.
specimen_type_concept_id	Yes	integer	A foreign key referring to the Concept identifier in the Standardized Vocabularies reflecting the system of record from which the Specimen was represented in the source data.
specimen_date	No	date	The date the specimen was obtained from the Person.
specimen_datetime	Yes	datetime	The date and time on the date when the Specimen was obtained from the person.
quantity	No	float	The amount of specimen collection from the person during the sampling procedure.
unit_concept_id	No	integer	A foreign key to a Standard Concept identifier for the Unit associated with the numeric quantity of the Specimen collection.
anatomic_site_concept_id	Yes	integer	A foreign key to a Standard Concept identifier for the anatomic location of specimen collection.
disease_status_concept_id	Yes	integer	A foreign key to a Standard Concept identifier for the Disease Status of specimen collection.
specimen_source_id	No	varchar(50)	The Specimen identifier as it appears in the source data.
specimen_source_value	No	varchar(50)	The Specimen value as it appears in the source data. This value is mapped to a Standard Concept in the Standardized Vocabularies and the original code is, stored here for reference.
unit_source_value	No	varchar(50)	The information about the Unit as detailed in the source.
anatomic_site_source_value	No	varchar(50)	The information about the anatomic site as detailed in the source.
disease_status_source_value	No	varchar(50)	The information about the disease status as detailed in the source.

7.2.1 Conventions

No.	Convention Description
1	Anatomic site is coded at the most specific level of granularity possible, such that higher level classifications can be derived using the Standardized Vocabularies.

7.3 FACT_RELATIONSHIP

The FACT_RELATIONSHIP table contains records about the relationships between facts stored as records in any table of the CDM. Relationships can be defined between facts from the same domain, or different domains. Examples of Fact Relationships include: Person relationships (parent-child), care site relationships (hierarchical organizational structure of facilities within a health system), indication relationship (between drug exposures and associated conditions), usage relationships (of devices during the course of an associated procedure), or facts derived from one another (measurements derived from an associated specimen).

Field	Required	Type	Description
domain_concept_id_1	Yes	integer	The concept representing the domain of fact one, from which the corresponding table can be inferred.
fact_id_1	Yes	integer	The unique identifier in the table corresponding to the domain of fact one.
domain_concept_id_2	Yes	integer	The concept representing the domain of fact two, from which the corresponding table can be inferred.
fact_id_2	Yes	integer	The unique identifier in the table corresponding to the domain of fact two.
relationship_concept_id	Yes	integer	A foreign key to a Standard Concept ID of relationship in the Standardized Vocabularies.

7.3.1 Conventions

No.	Convention Description

| 1 | All relationships are directional, and each relationship is represented twice symmetrically within the FACT_RELATIONSHIP table. For example, two persons if person_id = 1 is the mother of person_id = 2 two records are in the FACT_RELATIONSHIP table (all strings in fact concept_id records in the Concept table:

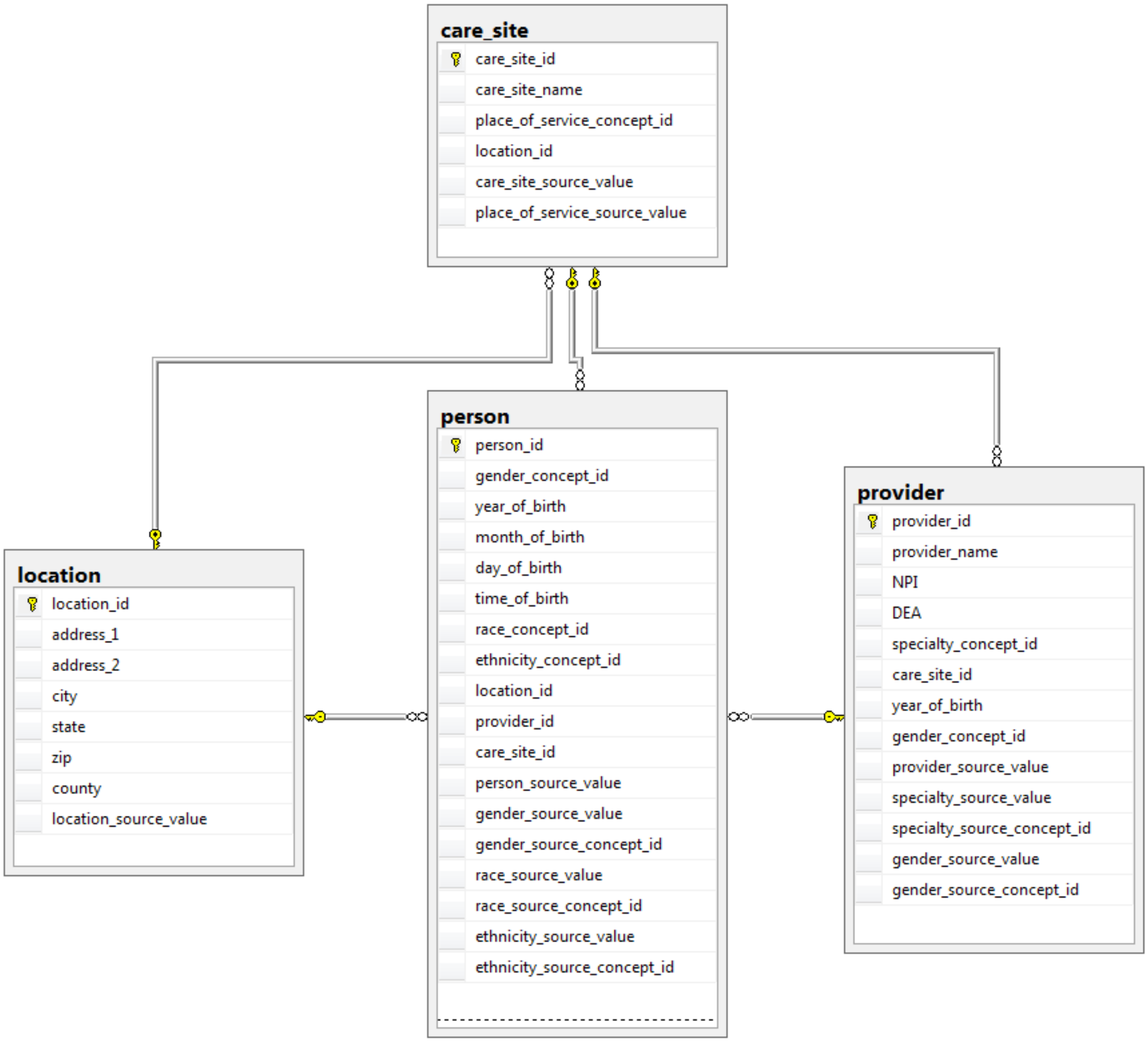
Person, 1, Person, 2, parent of

Person, 2, Person, 1, child of

8 Standardized Health System Data Tables

LOCATION LOCATION_HISTORY
CARE_SITE
PROVIDER

These tables describe the healthcare provider system responsible for administering the healthcare of the patient, rather than the demographic or clinical events the patient experienced. Below provides an entity-relationship diagram highlighting the tables within the Standardized Health System portion of the OMOP Common Data Model:



8.1 LOCATION

The LOCATION table represents a generic way to capture physical location or address information of Persons and Care Sites.

Field	Required	Type	Description
location_id	Yes	integer	A unique identifier for each geographic location.
address_1	No	varchar(50)	The address field 1, typically used for the street address, as it appears in the source data.

Field	Required	Type	Description
address_2	No	varchar(50)	The address field 2, typically used for additional detail such as buildings, suites, floors, as it appears in the source data.
city	No	varchar(50)	The city field as it appears in the source data.
state	No	varchar(2)	The state field as it appears in the source data.
zip	No	varchar(9)	The zip or postal code.
county	No	varchar(20)	The county.
country	No	varchar(100)	The country
location_source_value	No	varchar(50)	The verbatim information that is used to uniquely identify the location as it appears in the source data.
latitude	No	float	The geocoded latitude
longitude	No	float	The geocoded longitude

8.1.1 Conventions

No.	Convention Description
1	Each address or Location is unique and is present only once in the table.
2	Locations do not contain names, such as the name of a hospital. In order to construct a full address that can be used in the postal service, the address information from the Location needs to be combined with information from the Care Site. The PERSON table does not contain name information at all.
3	All fields in the Location tables contain the verbatim data in the source, no mapping or normalization takes place. None of the fields are mandatory. If the source data have no Location information at all, all Locations are represented by a single record. Typically, source data contain full or partial zip or postal codes or county or census district information.
4	Zip codes are handled as strings of up to 9 characters length. For US addresses, these represent either a 3-digit abbreviated Zip code as provided by many sources for patient protection reasons, the full 5-digit Zip or the 9-digit (ZIP + 4) codes. Unless for specific reasons analytical methods should expect and utilize only the first 3 digits. For international addresses, different rules apply.
5	The county information can be provided and is not redundant with information from the zip codes as not all of these have an unambiguous county designation.
6	For standardized geospatial visualization and analysis, addresses need to be, at the minimum be geocoded into latitude and longitude. This allows it to put as a point on a map. This proposal is to add two fields, latitude and longitude to the location table.

8.2 LOCATION_HISTORY

8.3 LOCATION_HISTORY

The LOCATION HISTORY table stores relationships between Persons or Care Sites and geographic locations over time.

Field	Required	Type	Description
location_id	Yes	integer	A foreign key to the location table.
relationship_type_concept_id	Yes	varchar(50)	The type of relationship between location and entity.
domain_id	Yes	varchar(50)	The domain of the entity that is related to the location. Either PERSON, PROVIDER, or CARE_SITE.
entity_id	Yes	integer	The unique identifier for the entity. References either person_id, provider_id, or care_site_id, depending on domain_id.
start_date	Yes	date	The date the relationship started.
end_date	No	date	The date the relationship ended.

8.3.1 Conventions

No.	Convention Description
1	The entities (and permissible domains) with related locations are: Persons (PERSON), Providers (PROVIDER), and Care Sites (CARE_SITE).
2	DOMAIN_ID specifies which table the ENTITY_ID refers to
3	Locations and entities are static. Relationships between locations and entities are dynamic.
4	When the domain is PERSON, the permissible values of relationship_type are: ‘residence’, ‘work site’, ‘school’.
5	When the domain is CARE_SITE, the value of relationship_type is NULL.
6	When the domain is PROVIDER, the value of relationship_type is ‘office’.

8.4 CARE_SITE

The CARE_SITE table contains a list of uniquely identified institutional (physical or organizational) units where healthcare delivery is practiced (offices, wards, hospitals, clinics, etc.).

Field	Required	Type	Description
care_site_id	Yes	integer	A unique identifier for each Care Site.
care_site_name	No	varchar(255)	The verbatim description or name of the Care Site as in data source

Field	Required	Type	Description
place_of_service_concept_id	Yes	integer	A foreign key that refers to a Place of Service Concept ID in the Standardized Vocabularies.
location_id	No	integer	A foreign key to the geographic Location in the LOCATION table, where the detailed address information is stored.
care_site_source_value	No	varchar(50)	The identifier for the Care Site in the source data, stored here for reference.
place_of_service_source_value	No	varchar(50)	The source code for the Place of Service as it appears in the source data, stored here for reference.

8.4.1 Conventions

No.	Convention Description
1	Care site is a unique combination of location_id and place_of_service_source_value.
2	Every record in the visit_occurrence table may have only one care site.
3	Care site does not take into account the provider (human) information such a specialty.
4	Many source data do not make a distinction between individual and institutional providers. The CARE_SITE table contains the institutional providers.
5	If the source, instead of uniquely identifying individual Care Sites, only provides limited information such as Place of Service, generic or "pooled" Care Site records are listed in the CARE_SITE table.
6	There are hierarchical and business relationships between Care Sites. For example, wards can belong to clinics or departments, which can in turn belong to hospitals, which in turn can belong to hospital systems, which in turn can belong to HMOs.
7	The relationships between Care Sites are defined in the FACT_RELATIONSHIP table.
8	The Care Site Source Value typically contains the name of the Care Site.
9	The Place of Service Concepts belongs to the Domain 'Place of Service'.

8.5 PROVIDER

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

Field	Required	Type	Description
provider_id	Yes	integer	A unique identifier for each Provider.
provider_name	No	varchar(255)	A description of the Provider.
npi	No	varchar(20)	The National Provider Identifier (NPI) of the provider.
dea	No	varchar(20)	The Drug Enforcement Administration (DEA) number of the provider.
specialty_concept_id	Yes	integer	A foreign key to a Standard Specialty Concept ID in the Standardized Vocabularies.
care_site_id	No	integer	A foreign key to the main Care Site where the provider is practicing.
year_of_birth	No	integer	The year of birth of the Provider.
gender_concept_id	Yes	integer	The gender of the Provider.
provider_source_value	No	varchar(50)	The identifier used for the Provider in the source data, stored here for reference.
specialty_source_value	No	varchar(50)	The source code for the Provider specialty as it appears in the source data, stored here for reference.
specialty_source_concept_id	Yes	integer	A foreign key to a Concept that refers to the code used in the source.
gender_source_value	No	varchar(50)	The gender code for the Provider as it appears in the source data, stored here for reference.
gender_source_concept_id	Yes	integer	A foreign key to a Concept that refers to the code used in the source.

8.5.1 Conventions

No.	Convention Description
1	Many sources do not make a distinction between individual and institutional providers. The PROVIDER table contains the individual providers.
2	If the source, instead of uniquely identifying individual providers, only provides limited information such as specialty, generic or 'pooled' Provider records are listed in the PROVIDER table.
3	A single Provider cannot be listed twice (be duplicated) in the table. If a Provider has more than one Specialty, the main or most often exerted specialty should be recorded.
4	Valid Specialty Concepts belong to the 'Specialty' domain.

No.	Convention Description
5	The CARE_SITE_ID represent a fixed relationship between a Provider and her main Care Site. Providers are also linked to Care Sites through Condition, Procedure and Visit records.

9 Standardized Health Economics Data Tables

PAYER_PLAN_PERIOD
COST

These tables contain cost information about healthcare. They are dependent on the healthcare delivery system the patient is involved in, which may vary significantly within a country and across different countries. However, the current model is focused on the US healthcare system.

9.1 PAYER_PLAN_PERIOD

The PAYER_PLAN_PERIOD table captures details of the period of time that a Person is continuously enrolled under a specific health Plan benefit structure from a given Payer. Each Person receiving healthcare is typically covered by a health benefit plan, which pays for (fully or partially), or directly provides, the care. These benefit plans are provided by payers, such as health insurances or state or government agencies. In each plan the details of the health benefits are defined for the Person or her family, and the health benefit Plan might change over time typically with increasing utilization (reaching certain cost thresholds such as deductibles), plan availability and purchasing choices of the Person. The unique combinations of Payer organizations, health benefit Plans and time periods in which they are valid for a Person are recorded in this table.

Field	Required	Type	Description
payer_plan_period_id	Yes	integer	A identifier for each unique combination of payer, plan, family code and time span.
person_id	Yes	integer	A foreign key identifier to the Person covered by the payer. The demographic details of that Person are stored in the PERSON table.
contract_person_id	No	integer	A foreign key identifier to the person_id in person table, for the person who is the primary subscriber/contract owner for the record in the payer_plan_period table. Maybe the same person or different person, depending on who is the primary subscriber/contract owner.
payer_plan_period_start_date	Yes	date	The start date of the payer plan period.
payer_plan_period_end_date	Yes	date	The end date of the payer plan period.
payer_concept_id	Yes	integer	A foreign key that refers to a standard Payer concept identifier in the Standarized Vocabularies
payer_source_value	No	varchar(50)	The source code for the payer as it appears in the source data.
payer_source_concept_id	Yes	integer	A foreign key to a payer concept that refers to the code used in the source.

Field	Required	Type	Description
plan_concept_id	Yes	integer	A foreign key that refers to a standard plan concept identifier that represents the health benefit plan in the Standardized Vocabularies.
plan_source_value	No	varchar(50)	The source code for the Person's health benefit plan as it appears in the source data.
plan_source_concept_id	Yes	integer	A foreign key to a plan concept that refers to the plan code used in the source data.
contract_concept_id	Yes	integer	A foreign key to a standard concept representing the reason justifying the contract between person_id and contract_person_id.
contract_source_value	No	integer	The source code representing the reason justifying the contract. Usually it is family relationship like a spouse, domestic partner, child etc.
contract_source_concept_id	Yes	integer	A foreign key to a concept that refers to the code used in the source as the reason justifying the contract.
sponsor_concept_id	Yes	integer	A foreign key that refers to a concept identifier that represents the sponsor in the Standardized Vocabularies.
sponsor_source_value	No	varchar(50)	The source code for the Person's sponsor of the health plan as it appears in the source data.
sponsor_source_concept_id	Yes	integer	A foreign key to a sponsor concept that refers to the sponsor code used in the source data.
family_source_value	No	varchar(50)	The source code for the Person's family as it appears in the source data.
stop_reason_concept_id	Yes	integer	A foreign key that refers to a standard termination reason that represents the reason for the termination in the Standardized Vocabularies.
stop_reason_source_value	No	varchar(50)	The reason for stop-coverage as it appears in the source data.
stop_reason_source_concept_id	Yes	integer	A foreign key to a stop-coverage concept that refers to the code used in the source.

9.1.1 Conventions

No.	Convention Description
1	Different Payers have different designs for their health benefit Plans. The PAYER_PLAN_PERIOD table does not capture all details of the plan design or the relationship between Plans or the cost of healthcare triggering a change from one Plan to another. However, it allows identifying the unique combination of Payer (insurer), Plan (determining healthcare benefits and limits) and Person. Typically, depending on healthcare utilization, a Person may have one or many subsequent Plans during coverage by a single Payer.
2	Typically, family members are covered under the same Plan as the Person. In those cases, the payer_source_value, plan_source_value and family_source_value are identical.
3	The contract_person_id is meant to refer to the owner of the plan, for instance, a parent who owns the plan under which the child is covered. Contract_person_id many times will be equal to person_id.

| 4 | The fields contract_source_value and contract_concept_id justify the contract relationship.

It is represented as the relationship from the person_id to contract_person_id. We will use SNOMED vocabulary of the Relationship Domain and Social Context concept class id (see here. For example:

Person_id is the spouse (4132413) of contract_person_id

person_id is the child (4285883) of the contract_person_id

5 | A patient can have multiple overlapping payer plan periods (THEMIS issue #18).

9.2 COST

The COST table captures records containing the cost of any medical event recorded in one of the OMOP clinical event tables such as DRUG_EXPOSURE, PROCEDURE_OCCURRENCE, VISIT_OCCURRENCE, VISIT_DETAIL, DEVICE_OCCURRENCE, OBSERVATION or MEASUREMENT.

Each record in the cost table account for the amount of money transacted for the clinical event. So, the COST table may be used to represent both receivables (charges) and payments (paid), each transaction type represented by its COST_CONCEPT_ID. The COST_TYPE_CONCEPT_ID field will use concepts in the Standardized Vocabularies to designate the source (provenance) of the cost data. A reference to the health plan information in the PAYER_PLAN_PERIOD table is stored in the record for information used for the adjudication system to determine the persons benefit for the clinical event.

Field	Required	Type	Description
cost_id	Yes	integer	A unique identifier for each COST record.
person_id	Yes	integer	A unique identifier for each PERSON.
cost_event_id	Yes	integer	A foreign key identifier to the event (e.g. Measurement, Procedure, Visit, Drug Exposure, etc) record for which cost data are recorded.

Field	Required	Type	Description
cost_event_field_concept_id	Yes	integer	A foreign key identifier to a concept in the CONCEPT table representing the identity of the field represented by COST_EVENT_ID
cost_concept_id	Yes	integer	A foreign key that refers to a Standard Cost Concept identifier in the Standardized Vocabularies belonging to the 'Cost' vocabulary.
cost_type_concept_id	Yes	integer	A foreign key identifier to a concept in the CONCEPT table for the provenance or the source of the COST data and belonging to the 'Cost Type' vocabulary
cost_source_concept_id	Yes	integer	A foreign key to a Cost Concept that refers to the code used in the source.
cost_source_value	No	varchar(50)	The source value for the cost as it appears in the source data
currency_concept_id	Yes	integer	A foreign key identifier to the concept representing the 3-letter code used to delineate international currencies, such as USD for US Dollar. These belong to the 'Currency' vocabulary
cost	Yes	float	The actual financial cost amount
incurred_date	Yes	date	The first date of service of the clinical event corresponding to the cost as in table capturing the information (e.g. date of visit, date of procedure, date of condition, date of drug etc).
billed_date	No	date	The date a bill was generated for a service or encounter
paid_date	No	date	The date payment was received for a service or encounter
revenue_code_concept_id	Yes	integer	A foreign key referring to a Standard Concept ID in the Standardized Vocabularies for Revenue codes belonging to the 'Revenue Code' vocabulary.
drg_concept_id	Yes	integer	A foreign key referring to a Standard Concept ID in the Standardized Vocabularies for DRG codes belonging to the 'DRG' vocabulary.
revenue_code_source_value	No	varchar(50)	The source value for the Revenue code as it appears in the source data, stored here for reference.
drg_source_value	No	varchar(50)	The source value for the 3-digit DRG source code as it appears in the source data, stored here for reference.
payer_plan_period_id	No	integer	A foreign key to the PAYER_PLAN_PERIOD table, where the details of the Payer, Plan and Family are stored. Record the payer_plan_id that relates to the payer who contributed to the paid_by_payer field.

9.2.1 Conventions

No.	Convention Description
1	The cost information is linked through the COST_EVENT_ID field to its entity, which denotes a record in a table referenced by the COST_EVENT_FIELD_CONCEPT_ID field.
2	One cost record is generated for each response by a payer. In a claims databases, the payment and payment terms reported by the payer for the goods or services billed will generate one cost record. If the source data has payment information for more than one payer (i.e. primary insurance and secondary insurance payment for one entity), then a cost record is created for each reporting payer. Therefore, it is possible for one procedure to have multiple cost records for each payer, but typically it contains one or no record per entity. Payer reimbursement cost records will be identified by using the PAYER_PLAN_ID field.
3	One cost record is generated for each money or currency amount associated with a record in one of the event tables.
4	The COST field represents the dollar amount, either incoming or outgoing
5	When dealing with summary costs, the cost of the goods or services the provider provides is often not known directly, but derived from the hospital charges multiplied by an average cost-to-charge ratio. This data is currently available for NIS datasets, or any other HCUP datasets. See also cost calculation explanation from AHRQ
7	Drug costs are composed of ingredient cost (the amount charged by the wholesale distributor or manufacturer), the dispensing fee (the amount charged by the pharmacy and the sales tax).
8	In claims data, generally there is one field representing the total payment from the payer for the service/device/drug. However, this field could be a calculated field if the source data provides separate payment information for the ingredient cost and the dispensing fee in case of prescription benefits. If there is more than one Payer in the source data, several cost records indicate that fact. The Payer reporting this reimbursement should be indicated under the PAYER_PLAN_ID field.
10	REVENUE_CODE_CONCEPT_ID: Revenue codes are a method to charge for a class of procedures and conditions in the U.S. hospital system.
11	DRG_CONCEPT_ID: Diagnosis Related Groups are US codes used to classify hospital cases into one of approximately 500 groups. Only the MS-DRG system should be used (mapped to vocabulary_id 'DRG) and all other DRG values should be mapped to 0 (THEMIS issue #19).

The COST table will store information reporting money or currency amounts. There are three types of cost data, defined in the COST_TYPE_CONCEPT_ID: 1) Payer is primary (coordination of benefit) 2) Payer is secondary (coordination of benefit) 3) Premium

One cost record is generated for each money or currency amount associated with a record in one of the event tables. For example, a raw record that looks like this:

patient_id	cob	coins	copay	deduct	net	total_pay	pddate	proc_cd	procmod	revcode	svcdate
175127601	0	0	22	3	88	113	2/28/2000	82378			1/1/2000

Will create four lines in the COST table:

cost_id	person_id	cost_event_id	cost_event_field_concept_id	cost_concept_id	cost_type_concept_id	currency_concept_id	cost	incurred_date	billed_date	paid_date	revenue_code_concept_id	drg_concept_id	cost_source_value	cost_source_concept_id	revenue_code_source_value	drg_source_value	payer_plan_period_id
1	175127601	1002	*TBD*	1234	5032	0	44818668	22	1/1/2000	2/28/2000	0	0					3045
2	175127601	1002	*TBD*	2345	5032	0	44818668	3	1/1/2000	2/28/2000	0	0					3045
3	175127601	1002	*TBD*	3456	5032	0	44818668	88	1/1/2000	2/28/2000	0	0					3045
4	175127601	1002	*TBD*	4567	5032	0	44818668	113	1/1/2000	2/28/2000	0	0					3045

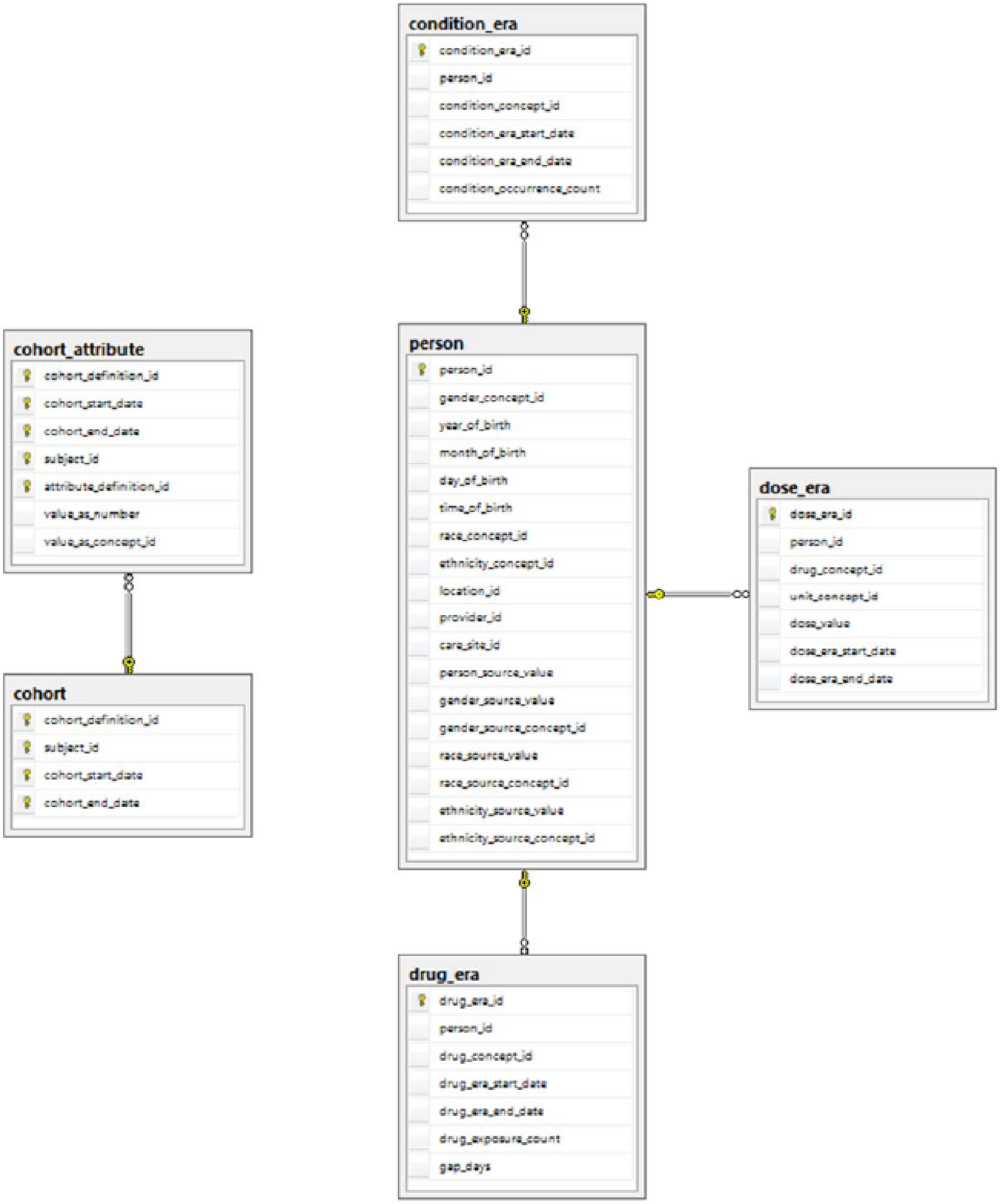
No.	Convention Description
1	The cost information is linked through the COST_EVENT_ID field to its entity, which denotes a record in a table referenced by the COST_EVENT_FIELD_CONCEPT_ID field.
2	One cost record is generated for each response by a payer. In a claims databases, the payment and payment terms reported by the payer for the goods or services billed will generate one cost record. If the source data has payment information for more than one payer (i.e. primary insurance and secondary insurance payment for one entity), then a cost record is created for each reporting payer. Therefore, it is possible for one procedure to have multiple cost records for each payer, but typically it contains one or no record per entity. Payer reimbursement cost records will be identified by using the PAYER_PLAN_ID field.
3	One cost record is generated for each money or currency amount associated with a record in one of the event tables.
4	The COST field represents the dollar amount, either incoming or outgoing
5	When dealing with summary costs, the cost of the goods or services the provider provides is often not known directly, but derived from the hospital charges multiplied by an average cost-to-charge ratio. This data is currently available for NIS datasets, or any other HCUP datasets. See also cost calculation explanation from AHRQ

No.	Convention Description
7	Drug costs are composed of ingredient cost (the amount charged by the wholesale distributor or manufacturer), the dispensing fee (the amount charged by the pharmacy and the sales tax).
8	In claims data, generally there is one field representing the total payment from the payer for the service/device/drug. However, this field could be a calculated field if the source data provides separate payment information for the ingredient cost and the dispensing fee in case of prescription benefits. If there is more than one Payer in the source data, several cost records indicate that fact. The Payer reporting this reimbursement should be indicated under the PAYER_PLAN_ID field.
10	REVENUE_CODE_CONCEPT_ID: Revenue codes are a method to charge for a class of procedures and conditions in the U.S. hospital system.
11	DRG_CONCEPT_ID: Diagnosis Related Groups are US codes used to classify hospital cases into one of approximately 500 groups. Only the MS-DRG system should be used (mapped to vocabulary_id 'DRG) and all other DRG values should be mapped to 0.

10 Standardized Derived Elements

DRUG_ERA
DOSE_ERA
CONDITION_ERA

These tables contain information about the clinical events of a patient that are not obtained directly from the raw source data, but from other tables of the CDM. Below provides an entity-relationship diagram highlighting the tables within the Standardized Derived Elements portion of the OMOP Common Data Model:



10.1 DRUG_ERA

A Drug Era is defined as a span of time when the Person is assumed to be exposed to a particular active ingredient. A Drug Era is not the same as a Drug Exposure: Exposures are individual records corresponding to the source when Drug was delivered to the Person, while successive periods of Drug Exposures are combined under certain rules to produce continuous Drug Eras.

Field	Required	Type	Description
drug_era_id	Yes	integer	A unique identifier for each Drug Era.
person_id	Yes	integer	A foreign key identifier to the Person who is subjected to the Drug during the fDrug Era. The demographic details of that Person are stored in the PERSON table.
drug_concept_id	Yes	integer	A foreign key that refers to a Standard Concept identifier in the Standardized Vocabularies for the Ingredient Concept.
drug_era_start_datetime	Yes	date	The start date for the Drug Era constructed from the individual instances of Drug Exposures. It is the start date of the very first chronologically recorded instance of conutilization of a Drug.
drug_era_end_datetime	Yes	date	The end date for the drug era constructed from the individual instance of drug exposures. It is the end date of the final continuously recorded instance of utilization of a drug.
drug_exposure_count	No	integer	The number of individual Drug Exposure occurrences used to construct the Drug Era.
gap_days	No	integer	The number of days that are not covered by DRUG_EXPOSURE records that were used to make up the era record.

10.1.1 Conventions

No.	Convention Description
1	Drug Eras are derived from records in the DRUG_EXPOSURE table using a standardized algorithm.
2	Each Drug Era corresponds to one or many Drug Exposures that form a continuous interval and contain the same Drug Ingredient (active compound).
3	The drug_concept_id field only contains Concepts that have the concept_class 'Ingredient'. The Ingredient is derived from the Drug Concepts in the DRUG_EXPOSURE table that are aggregated into the Drug Era record.
4	The Drug Era Start Date is the start date of the first Drug Exposure.

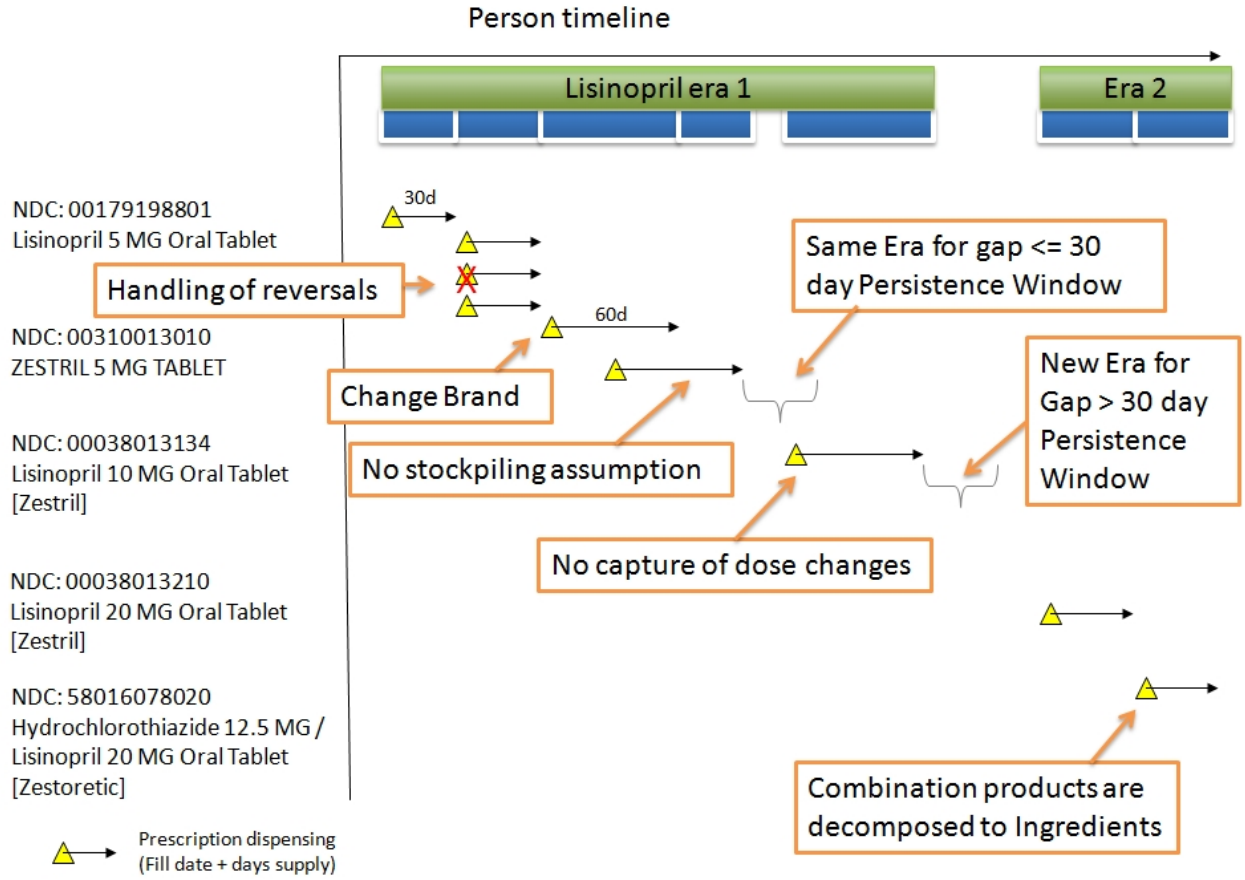
| 5 | The Drug Era End Date is the end date of the last Drug Exposure. The End Date of each Drug Exposure is either taken from the field drug_exposure_end_date or, as it is typically not available, inferred using the following rules:

For pharmacy prescription data, the date when the drug was dispensed plus the number of days of supply are used to extrapolate the End Date for the Drug Exposure. Depending on the country-specific healthcare system, this supply information is either explicitly provided in the day_supply field or inferred from package size or similar information.

For Procedure Drugs, usually the drug is administered on a single date (i.e., the administration date).

A standard Persistence Window of 30 days (gap, slack) is permitted between two subsequent such extrapolated DRUG_EXPOSURE records to be considered to be merged into a single Drug Era.

| | 6 | The Gap Days determine how many total drug-free days are observed between all Drug Exposure events that contribute to a DRUG_ERA record. It is assumed that the drugs are "not stockpiled" by the patient, i.e. that if a new drug prescription or refill is observed (a new DRUG_EXPOSURE record is written), the remaining supply from the previous events is abandoned. | | 7 | The difference between Persistence Window and Gap Days is that the former is the maximum drug-free time allowed between two subsequent DRUG_EXPOSURE records, while the latter is the sum of actual drug-free days for the given Drug Era under the above assumption of non-stockpiling. | | 8 | The choice of a standard Persistence Window of 30 and the non-stockpiling assumption is arbitrary, but has been shown to deliver good results in drug-outcome estimation. Other problems, such as estimation of drug compliance, my require a different or drug-dependent Persistence Window/stockpiling assumption. Researchers are encouraged to consider creating their own Drug Eras with different parameters as Cohorts and store them in the COHORT table. |


Person timeline
Lisinopril era 1
Era 2
NDC: 00179198801
Lisinopril 5 MG Oral Tablet
30d
Handling of reversals
60d
NDC: 00310013010
ZESTRIL 5 MG TABLET
Change Brand
Same Era for gap <= 30 day Persistence Window
NDC: 00038013134
Lisinopril 10 MG Oral Tablet [Zestril]
No stockpiling assumption
New Era for Gap > 30 day Persistence Window
No capture of dose changes
NDC: 00038013210
Lisinopril 20 MG Oral Tablet [Zestril]
NDC: 58016078020
Hydrochlorothiazide 12.5 MG / Lisinopril 20 MG Oral Tablet [Zestoretic]
Combination products are decomposed to Ingredients
Prescription dispensing
(Fill date + days supply)

10.2 DOSE_ERA

A Dose Era is defined as a span of time when the Person is assumed to be exposed to a constant dose of a specific active ingredient.

Field	Required	Type	Description
dose_era_id	Yes	integer	A unique identifier for each Dose Era.
person_id	Yes	integer	A foreign key identifier to the Person who is subjected to the drug during the drug era. The demographic details of that Person are stored in the PERSON table.
drug_concept_id	Yes	integer	A foreign key that refers to a Standard Concept identifier in the Standardized Vocabularies for the active Ingredient Concept.
unit_concept_id	Yes	integer	A foreign key that refers to a Standard Concept identifier in the Standardized Vocabularies for the unit concept.
dose_value	Yes	float	The numeric value of the dose.
dose_era_start_datetime	Yes	date	The start date for the drug era constructed from the individual instances of drug exposures. It is the start date of the very first chronologically recorded instance of utilization of a drug.
dose_era_end_datetime	Yes	date	The end date for the drug era constructed from the individual instance of drug exposures. It is the end date of the final continuously recorded instance of utilization of a drug.

10.2.1 Conventions

No.	Convention Description
1	Dose Eras will be derived from records in the DRUG_EXPOSURE table and the Dose information from the DRUG_STRENGTH table using a standardized algorithm.
2	Each Dose Era corresponds to one or many Drug Exposures that form a continuous interval and contain the same Drug Ingredient (active compound) at the same effective daily dose.
3	Dose Form information is not taken into account. So, if the patient changes between different formulations, or different manufacturers with the same formulation, the Dose Era is still spanning the entire time of exposure to the Ingredient.

No.	Convention Description
4	The daily dose is calculated for each DRUG_EXPOSURE record by calculating the total dose of the record and dividing by the duration.

The total dose of a DRUG_EXPOSURE record is calculated with the help of the DRUG_STRENGTH table containing the dosage information for each drug as following:

5	Tablets and other fixed amount formulations
	Example: Acetaminophen (Paracetamol) 500 mg, 20 tablets.
DRUG_STRENGTH	The denominator_unit is empty
DRUG_EXPOSURE	The quantity refers to number of pieces, e.g. tablets *In the example: 20*

`Ingredient dose=`	`quantity x amount_value [amount_unit_concept_id]`
	`Acetaminophen dose = 20 x 500mg = 10,000mg`

6	Puffs of an inhaler
	Note: There is no difference to use case 1 besides that the DRUG_STRENGTH table may put {actuat} in the denominator unit. In this case the strength is provided in the numerator.
DRUG_STRENGTH	The denominator_unit is {actuat}
DRUG_EXPOSURE	The quantity refers to the number of pieces, e.g. puffs
`Ingredient dose=`	`quantity x numerator_value [numerator_unit_concept_id]`

7	Quantified Drugs which are formulated as a concentration
	Example: The Clinical Drug is Acetaminophen 250 mg/mL in a 5mL oral suspension. The Quantified Clinical Drug would have 1250 mg / 5 ml in the DRUG_STRENGTH table. Two suspensions are dispensed.
DRUG_STRENGTH	The denominator_unit is either mg or mL. The denominator_value might be different from 1.
DRUG_EXPOSURE	The quantity refers to a fraction or, multiple of the pack. *Example: 2*

`Ingredient dose=`	`quantity x numerator_value [numerator_unit_concept_id]`
	`Acetaminophen dose = 2 x 1250mg = 2500mg`

8	Drugs with the total amount provided in quantity, e.g. chemotherapeutics
	Example: 42799258 “Benzyl Alcohol 0.1 ML/ML / Pramoxine hydrochloride 0.01 MG/MG Topical Gel” dispensed in a 1.25oz pack.
DRUG_STRENGTH	The denominator_unit is either mg or mL. *Example: Benzyl Alcohol in mL and Pramoxine hydrochloride in mg*

8	Drugs with the total amount provided in quantity, e.g. chemotherapeutics
DRUG_EXPOSURE	The quantity refers to mL or g. *Example: 1.25 x 30 (conversion factor oz -> mL) = 37*
`Ingredient dose=`	`quantity x numerator_value [numerator_unit_concept_id]` *Benzyl Alcohol dose = 37 x 0.1mL = 3.7mL* *Pramoxine hydrochloride dose = 37 x 0.01mg x 1000 = 370mg* *Note: The analytical side should check the denominator in the DRUG_STRENGTH table. As mg is used for the second ingredient the factor 1000 will be applied to convert between g and mg.*

9	Compounded drugs
	Example: Ibuprofen 20%/Piroxicam 1% Cream, 30ml in 5ml tubes.
DRUG_STRENGTH	We need entries for the ingredients of Ibuprofen and Piroxicam, probably with an amount_value of 1 and a unit of mg.
DRUG_EXPOSURE	The quantity refers to the total amount of the compound. Use one record in the DRUG_EXPOSURE table for each compound. *Example: 20% Ibuprofen of 30ml = 6mL, 1% Piroxicam of 30ml = 0.3mL*
`Ingredient dose=`	Depends on the drugs involved: One of the use cases above. *Ibuprofen dose = 6 x 1mg x 1000 = 6000mg* *Piroxicam dose = 0.3 x 1mg x 1000 = 300mg* *Note: The analytical side determines that the denominator for both ingredients in the DRUG_STRENGTH table is mg and applies the factor 1000 to convert between mL/g and mg.*

10	Drugs with the active ingredient released over time, e.g. patches
	Example: Ethinyl Estradiol 0.000833 MG/HR / norelgestromin 0.00625 MG/HR Weekly Transdermal Patch
DRUG_STRENGTH	The denominator units refer to hour. *Example: Ethinyl Estradiol 0.000833 mg/h / norelgestromin 0.00625 mg/h*
DRUG_EXPOSURE	The quantity refers to the number of pieces. *Example: 1 patch*
`Ingredient rate=`	`numerator_value [numerator_unit_concept_id]` *Ethinyl Estradiol rate = 0.000833 mg/h* *norelgestromin rate 0.00625 mg/h* *Note: This can be converted to a daily dosage by multiplying it with 24. (Assuming 1 patch at a time for at least 24 hours)*

10.3 CONDITION_ERA

A Condition Era is defined as a span of time when the Person is assumed to have a given condition. Similar to Drug Eras, Condition Eras are chronological periods of Condition Occurrence. Combining individual Condition Occurrences into a single Condition Era serves two purposes:

- It allows aggregation of chronic conditions that require frequent ongoing care, instead of treating each Condition Occurrence as an independent event.
- It allows aggregation of multiple, closely timed doctor visits for the same Condition to avoid double-counting the Condition Occurrences.

For example, consider a Person who visits her Primary Care Physician (PCP) and who is referred to a specialist. At a later time, the Person visits the specialist, who confirms the PCP's original diagnosis and provides the appropriate treatment to resolve the condition. These two independent doctor visits should be aggregated into one Condition Era.

Field	Required	Type	Description
condition_era_id	Yes	integer	A unique identifier for each Condition Era.
person_id	Yes	integer	A foreign key identifier to the Person who is experiencing the Condition during the Condition Era. The demographic details of that Person are stored in the PERSON table.
condition_concept_id	Yes	integer	A foreign key that refers to a standard Condition Concept identifier in the Standardized Vocabularies.
condition_era_start_datetime	Yes	date	The start date for the Condition Era constructed from the individual instances of Condition Occurrences. It is the start date of the very first chronologically recorded instance of the condition.
condition_era_end_datetime	Yes	date	The end date for the Condition Era constructed from the individual instances of Condition Occurrences. It is the end date of the final continuously recorded instance of the Condition.
condition_occurrence_count	No	integer	The number of individual Condition Occurrences used to construct the condition era.

10.3.1 Conventions

No.	Convention Description
1	Condition Era records will be derived from the records in the CONDITION_OCCURRENCE table using a standardized algorithm.

| 2 | Each Condition Era corresponds to one or many Condition Occurrence records that form a continuous interval.

The condition_concept_id field contains Concepts that are identical to those of the CONDITION_OCCURRENCE table records that make up the Condition Era. In contrast to Drug Eras, Condition Eras are not aggregated to contain Conditions of different hierarchical layers.

The Condition Era Start Date is the start date of the first Condition Occurrence.

The Condition Era End Date is the end date of the last Condition Occurrence.

3 | Condition Eras are built with a Persistence Window of 30 days, meaning, if no occurrence of the same condition_concept_id happens within 30 days of any one occurrence, it will be considered the condition_era_end_date.

11 Results Schema

COHORT
COHORT_DEFINITION

New to CDM v6.0 is the concept of schemas. This allows for more separation between read-only and writeable tables. The clinical data, event, and vocabulary tables are in the 'CDM' schema and tables that need to be manipulated by web-based tools or end users have moved to the 'Results' schema. Currently the only two tables in the 'Results' schema are COHORT and COHORT_DEFINITON, though likely more will be added over the course of v6.0 point releases.

11.1 COHORT

The COHORT table contains records of subjects that satisfy a given set of criteria for a duration of time. The definition of the cohort is contained within the COHORT_DEFINITION table. Cohorts can be constructed of patients (Persons), Providers or Visits.

Field	Required	Type	Description
cohort_definition_id	Yes	integer	A foreign key to a record in the COHORT_DEFINITION table containing relevant Cohort Definition information.
subject_id	Yes	integer	A foreign key to the subject in the cohort. These could be referring to records in the PERSON, PROVIDER, VISIT_OCCURRENCE table.
cohort_start_date	Yes	date	The date when the Cohort Definition criteria for the Person, Provider or Visit first match.
cohort_end_date	Yes	date	The date when the Cohort Definition criteria for the Person, Provider or Visit no longer match or the Cohort membership was terminated.

11.1.1 Conventions

- The core of a Cohort is the unifying definition or feature of the Cohort. This is captured in the cohort_definition_id. For example, Cohorts can include patients diagnosed with a specific condition, patients exposed to a particular drug, or Providers who have performed a specific Procedure.
- Cohort records must have a Start Date
- Cohort records must have an End Date, but may be set to Start Date or could have applied a censored date using the Observation Period Start Date.
- Cohort records must contain a Subject Id, which can refer to the Person, Provider, Visit record or Care Site. The Cohort Definition will define the type of subject through the subject concept id.
- A subject can belong (or not belong) to a cohort at any moment in time
- A subject can only have one record in the cohort table for any moment of time, i.e. it is not possible for a person to contain multiple records indicating cohort membership that are overlapping in time

11.2 COHORT_DEFINITION

The COHORT_DEFINITION table contains records defining a Cohort derived from the data through the associated description and syntax and upon instantiation (execution of the algorithm) placed into the COHORT table. Cohorts are a set of subjects that satisfy a given combination of inclusion criteria for a duration of time. The COHORT_DEFINITION table provides a standardized structure for maintaining the rules governing the inclusion of a subject into a cohort, and can store operational programming code to instantiate the cohort within the OMOP Common Data Model.

Field	Required	Type	Description
cohort_definition_id	Yes	integer	A unique identifier for each Cohort.
cohort_definition_name	Yes	varchar(255)	A short description of the Cohort.
cohort_definition_description	No	varchar(MAX)	A complete description of the Cohort definition
definition_type_concept_id	Yes	integer	Type defining what kind of Cohort Definition the record represents and how the syntax may be executed
cohort_definition_syntax	No	varchar(MAX)	Syntax or code to operationalize the Cohort definition
subject_concept_id	Yes	integer	A foreign key to the Concept to which defines the domain of subjects that are members of the cohort (e.g., Person, Provider, Visit).
cohort_initiation_date	No	Date	A date to indicate when the Cohort was initiated in the COHORT table

11.2.1 Conventions

- The cohort_definition_syntax does not prescribe any specific syntax or programming language. Typically, it would be any flavor SQL, a cohort definition language, or a free-text description of the algorithm.
- The subject_concept_id determines what the individual subjects or entities of the Cohort consists of. In most cases, that would be a Person (patient). But cohorts could also be constructed for Providers, Visits or any other Domain. Note that the Domain is not codified using the alphanumerical domain_id like in the CONCEPT table. Instead, the corresponding Concept is used. The Concepts for each domain can be obtained from the DOMAIN table in the domain_concept_id.