

# OMOP Common Data Model Specifications

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

*2018-06-14*

## Contents

<b>1</b>	<b>License</b>	<b>2</b>
<b>2</b>	<b>Background</b>	<b>2</b>
2.1	The Role of the Common Data Model . . . . .	3
2.2	Design Principles . . . . .	3
2.3	Data Model Conventions . . . . .	4
<b>3</b>	<b>Glossary of Terms</b>	<b>7</b>
<b>4</b>	<b>Standardized Vocabularies</b>	<b>9</b>
4.1	CONCEPT . . . . .	10
4.2	VOCABULARY . . . . .	14
4.3	DOMAIN . . . . .	16
4.4	CONCEPT_CLASS . . . . .	16
4.5	CONCEPT_RELATIONSHIP . . . . .	19
4.6	RELATIONSHIP . . . . .	20
4.7	CONCEPT_SYNONYM . . . . .	26
4.8	CONCEPT_ANCESTOR . . . . .	27
4.9	SOURCE_TO_CONCEPT_MAP . . . . .	28
4.10	DRUG_STRENGTH . . . . .	29
4.11	COHORT_DEFINITION . . . . .	31
4.12	ATTRIBUTE_DEFINITION . . . . .	31
<b>5</b>	<b>Standardized Metadata</b>	<b>32</b>
5.1	CDM_SOURCE . . . . .	33
<b>6</b>	<b>Standardized Clinical Data Tables</b>	<b>34</b>
6.1	PERSON . . . . .	35
6.2	OBSERVATION_PERIOD . . . . .	37
6.3	SPECIMEN . . . . .	38
6.4	DEATH . . . . .	39
6.5	VISIT_OCCURRENCE . . . . .	40
6.6	VISIT_DETAIL . . . . .	41
6.7	PROCEDURE_OCCURRENCE . . . . .	43
6.8	DRUG_EXPOSURE . . . . .	45
6.9	DEVICE_EXPOSURE . . . . .	47
6.10	CONDITION_OCCURRENCE . . . . .	49
6.11	MEASUREMENT . . . . .	50
6.12	NOTE . . . . .	53
6.13	NOTE_NLP . . . . .	56
6.14	OBSERVATION . . . . .	58
6.15	FACT_RELATIONSHIP . . . . .	60
<b>7</b>	<b>Standardized Health System Data Tables</b>	<b>60</b>
7.1	LOCATION . . . . .	61
7.2	CARE_SITE . . . . .	62

7.3 PROVIDER . . . . .	63
<b>8 Standardized Health Economics Data Tables</b>	<b>64</b>
8.1 PAYER_PLAN_PERIOD . . . . .	64
8.2 COST . . . . .	65
<b>9 Standardized Derived Elements</b>	<b>69</b>
9.1 COHORT . . . . .	70
9.2 COHORT_ATTRIBUTE . . . . .	71
9.3 DRUG_ERA . . . . .	72
9.4 DOSE_ERA . . . . .	74
9.5 CONDITION_ERA . . . . .	77

## 1 License

© 2014 Observational Health Data Sciences and Informatics

This work is based on work by the Observational Medical Outcomes Partnership (OMOP) and used under license from the FNIH at <http://omop.fnih.org/publiclicense>.

All derivative work after the OMOP CDM v4 specification is dedicated to the public domain. Observational Health Data Sciences and Informatics (OHDSI) has waived all copyright and related or neighboring rights to the extent allowed by law.



<http://creativecommons.org/publicdomain/zero/1.0/>

## 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 analyzing disparate data sources across the health care delivery spectrum, and
- 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](http://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 at providing data organized in a way optimal for analysis, rather than for the purpose of 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. Older versions of the CDM can be easily created from this 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 methods against the CDM need to understand these conventions.

### 2.3.1 General conventions of data tables

The CDM is platform-independent. Data types are defined generically using ANSI SQL data types (VARCHAR, INTEGER, FLOAT, DATE, TIME, 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 time format. Standard queries against CDM may vary for local instantiations and date/time 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.

### 2.3.2 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
_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 reference value for 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 = 35708202 denotes the concept of ‘Nausea’ in the MedDRA terminology; the analogous condition_concept_id might be 31967, since SNOMED-CT is the Standardized Vocabularies 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’

### 2.3.3 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 unmappable concept.

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

### 2.3.4 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 contains the codes from public code systems such as ICD-9-CM, NDC, CPT-4 etc. or local controlled vocabularies (such as F for female and M for male) copied from the source data. Source Values are stored in the `_source_value` field 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, but 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.

### 2.3.5 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).

### 2.3.6 Time span of available data

Data tables for clinical data contain a date stamp (ending in `_date`, `_start_date` or `_end_date`), 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 happened prior the first OBSERVATION\_PERIOD, it will be captured as a record in the OBSERVATION table of ‘Medical history’ (`concept_id` = 43054928), with the `observation_date` 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.

### 2.3.7 Content of each table

For the tables of the main domains of the CDM it is imperative that used concepts 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: V25.5 “Encounter for insertion of implantable subdermal contraceptive” defines a procedure and is therefore stored in the `PROCEDURE_OCCURRENCE` table.

### 2.3.8 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, are stored as unstructured text, and are generally not to be used by any standardized analytics.
- 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 directly to standard `concept_ids`. Store these mappings in the `SOURCE_TO_CONCEPT_MAP` table.
  - If the source code is not mappable to a vocabulary term, the `source_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:

- Map all your source values to the corresponding `concept_ids` in the source vocabulary. 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 mappable to a vocabulary term, 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_id` 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, HCPCS code for infusion of a drug will map to a concept in the procedure domain of the infusion and a different concept in the drug domain for the product infused. It is also possible for one `source_concept_id` to map to multiple standard `concept_ids` within the same domain. For example, ICD-9 for ‘viral hepatitis with hepatic coma’ maps to SNOMED ‘viral hepatitis’ and a different 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 table(s) corresponding to the domain of the standard `concept_id(s)`.
  - If the source value is mapped to `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 its 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 unmappable `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).

There is no constraint on allowed `concept_ids` within the `source_concept_id` fields.

### 2.3.9 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.

Term	Abbr.	Description
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.
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.



Term	Abbr.	Description
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).
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  
COHORT\_DEFINITION  
ATTRIBUTE\_DEFINITION

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

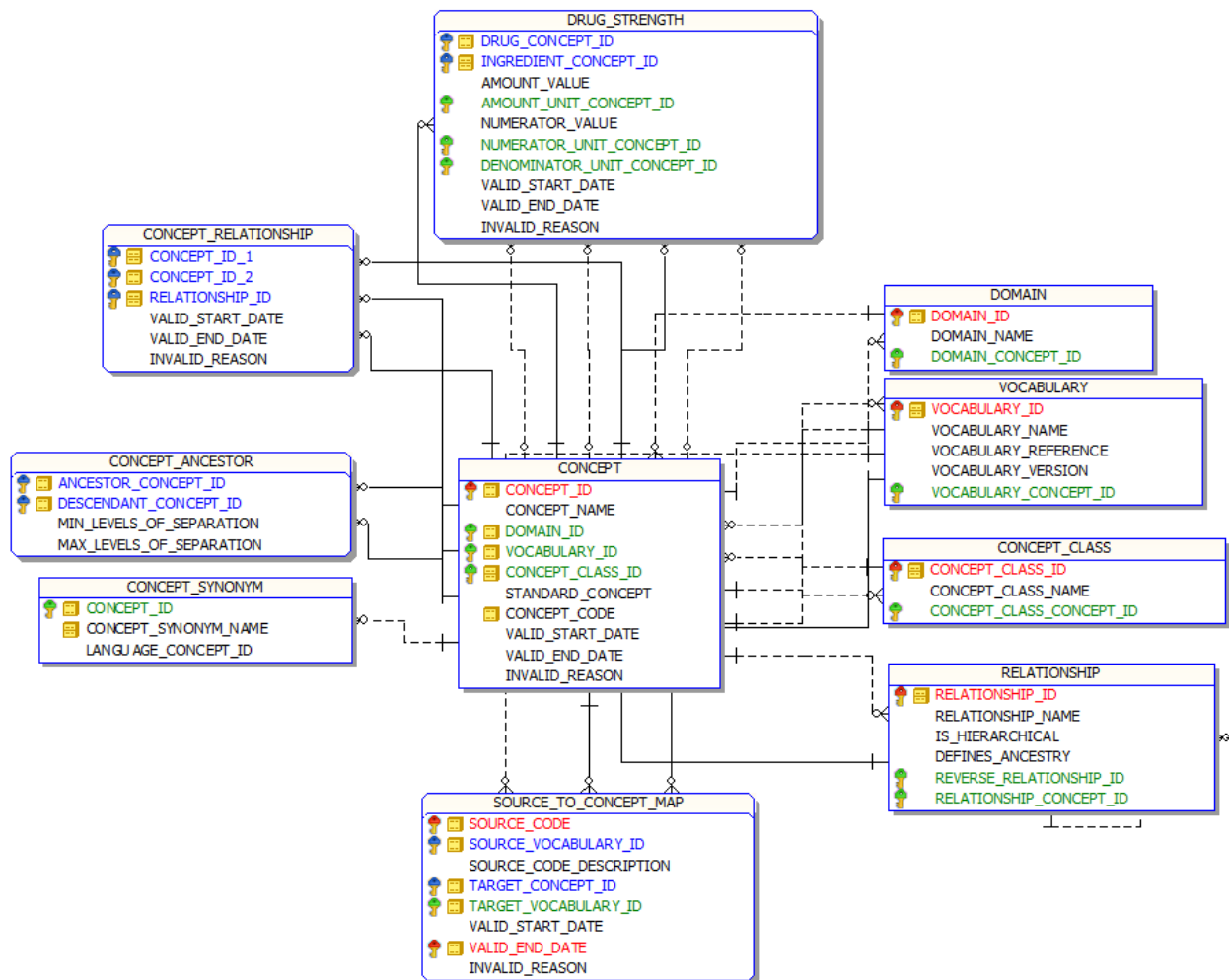


Figure 1: Vocabulary entity-relationship diagram

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.
domain_id	Yes	varchar(20)	A foreign key to the <a href="#">DOMAIN</a> table the Concept belongs to.
vocabulary_id	Yes	varchar(20)	A foreign key to the <a href="#">VOCABULARY</a> 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 allowable 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:

- All Concepts are maintained centrally by the CDM and Vocabularies Working Group. Additional concepts can be added, as needed, upon request.
- 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.
- The `concept_id` of a Concept is persistent, i.e. stays the same for the same Concept between releases of the Standardized Vocabularies.
- 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.
- Each Concept is assigned to a Domain. For Standard Concepts, there 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.
- For details of the Vocabularies adopted for use in the OMOP CDM refer to the Standardized Vocabularies [specification](#).
- 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.
- 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.
- 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.
- 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.
- 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, there is no meaningful `concept_code`.
- 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 .
- Concepts without a known instantiated date are assigned `valid_start_date` of ‘1-Jan-1970’.
- Concepts that are not invalid are assigned `valid_end_date` of ‘31-Dec-2099’.
- 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.
- 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.

## 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	Yes	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

- 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.
- The vocabulary\_id field contains an alphanumeric identifier, that can also be used as the abbreviation of the Vocabulary name.
- The record with vocabulary\_id = ‘None’ is reserved to contain information regarding the current version of the Entire Standardized Vocabularies.
- The vocabulary\_name field contains the full official name of the Vocabulary, as well as the source or vendor in parenthesis.
- 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.
- In past versions of the VOCABULARY table, the vocabulary\_id used to be a numerical value. A conversion table between these old and new IDs is given below:

Previous VOCABULARY_ID	Version 5 VOCABULARY_ID
0	None
1	<a href="#">SNOMED</a>

Previous VOCABULARY_ID	Version 5 VOCABULARY_ID
2	ICD9CM
3	ICD9Proc
4	CPT4
5	HCPCS
6	LOINC
7	NDFRT
8	RxNorm
9	NDC
10	GPI
11	UCUM
12	Gender
13	Race
14	Place of Service
15	MedDRA
16	Multum
17	Read
18	OXMIS
19	Indication
20	ETC
21	ATC
22	Multilex
24	Visit
28	VA Product
31	SMQ
32	VA Class
33	Cohort
34	ICD10
35	ICD10PCS
36	Drug Type
37	Condition Type
38	Procedure Type
39	Observation Type
40	DRG
41	MDC
42	APC
43	Revenue Code
44	Ethnicity
45	Death Type
46	Mesh
47	NUCC
48	Specialty
49	LOINC
50	SPL
53	Genseqno
54	CCS
55	OPCS4
56	Gemscript
57	HES Specialty
58	Note Type
59	Domain
60	PCORNet
61	Obs Period Type

Previous VOCABULARY_ID	Version 5 VOCABULARY_ID
62	Visit Type
63	Device Type
64	Meas Type
65	Currency
66	Relationship
67	Vocabulary
68	Concept Class
69	Cohort Type
70	ICD10CM

### 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
<code>domain_id</code>	Yes	<code>varchar(20)</code>	A unique key for each domain.
<code>domain_name</code>	Yes	<code>varchar(255)</code>	The name describing the Domain, e.g. “Condition”, “Procedure”, “Measurement” etc.
<code>domain_concept_id</code>	Yes	<code>integer</code>	A foreign key that refers to an identifier in the <code>CONCEPT</code> table for the unique Domain Concept the Domain record belongs to.

#### 4.3.1 Conventions

- 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.
- The `domain_id` field contains an alphanumeric identifier, that can also be used as the abbreviation of the Domain.
- The `domain_name` field contains the unabbreviated names of the Domain.
- 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 Concept.
- Versions prior to v5.0.0 of the OMOP CDM did not support the notion of a Domain.

### 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 <a href="#">CONCEPT</a> table for the unique Concept Class the record belongs to.

#### 4.4.1 Conventions

- 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.
- The concept\_class\_id field contains an alphanumeric identifier, that can also be used as the abbreviation of the Concept Class.
- The concept\_class\_name field contains the unabbreviated names of the Concept Class.
- 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.
- Past versions of the OMOP CDM did not have a separate reference table for all Concept Classes. Also, the content of the old concept\_class and the new concept\_class\_id fields are not always identical. A conversion table can be found here:

Previous CONCEPT_CLASS	Version 5 CONCEPT_CLASS_ID
Administrative concept	Admin Concept
Admitting Source	Admitting Source
Anatomical Therapeutic Chemical Classification	ATC
Anatomical Therapeutic Chemical Classification	ATC
APC	Procedure
Attribute	Attribute
Biobank Flag	Biobank Flag
Biological function	Biological Function
Body structure	Body Structure
Brand Name	Brand Name
Branded Drug	Branded Drug
Branded Drug Component	Branded Drug Comp
Branded Drug Form	Branded Drug Form
Branded Pack	Branded Pack
CCS_DIAGNOSIS	Condition
CCS_PROCEDURES	Procedure
Chart Availability	Chart Availability
Chemical Structure	Chemical Structure
Clinical Drug	Clinical Drug
Clinical Drug Component	Clinical Drug Comp
Clinical Drug Form	Clinical Drug Form
Clinical finding	Clinical Finding

Previous CONCEPT_CLASS	Version 5 CONCEPT_CLASS_ID
Clinical Pack	Clinical Pack
Concept Relationship	Concept Relationship
Condition Occurrence Type	Condition Occur Type
Context-dependent category	Context-dependent
CPT-4	Procedure
Currency	Currency
Death Type	Death Type
Device Type	Device Type
Discharge Disposition	Discharge Dispo
Discharge Status	Discharge Status
Domain	Domain
Dose Form	Dose Form
DRG	Diagnostic Category
Drug Exposure Type	Drug Exposure Type
Drug Interaction	Drug Interaction
Encounter Type	Encounter Type
Enhanced Therapeutic Classification	ETC
Enrollment Basis	Enrollment Basis
Environment or geographical location	Location
Ethnicity	Ethnicity
Event	Event
Gender	Gender
HCPCS	Procedure
Health Care Provider Specialty	Provider Specialty
HES specialty	Provider Specialty
High Level Group Term	HLGT
High Level Term	HLT
Hispanic	Hispanic
ICD-9-Procedure	Procedure
Indication or Contra-indication	Ind / CI
Ingredient	Ingredient
LOINC Code	Measurement
LOINC Multidimensional Classification	Meas Class
Lowest Level Term	LLT
MDC	Diagnostic Category
Measurement Type	Meas Type
Mechanism of Action	Mechanism of Action
Model component	Model Comp
Morphologic abnormality	Morph Abnormality
MS-DRG	Diagnostic Category
Namespace concept	Namespace Concept
Note Type	Note Type
Observable entity	Observable Entity
Observation Period Type	Obs Period Type
Observation Type	Observation Type
OMOP DOI cohort	Drug Cohort
OMOP HOI cohort	Condition Cohort
OPCS-4	Procedure
Organism	Organism
Patient Status	Patient Status
Pharmaceutical / biologic product	Pharma/Biol Product
Pharmaceutical Preparations	Pharma Preparation

Previous CONCEPT_CLASS	Version 5 CONCEPT_CLASS_ID
Pharmacokinetics	PK
Pharmacologic Class	Pharmacologic Class
Physical force	Physical Force
Physical object	Physical Object
Physiologic Effect	Physiologic Effect
Place of Service	Place of Service
Preferred Term	PT
Procedure	Procedure
Procedure Occurrence Type	Procedure Occur Type
Qualifier value	Qualifier Value
Race	Race
Record artifact	Record Artifact
Revenue Code	Revenue Code
Sex	Gender
Social context	Social Context
Special concept	Special Concept
Specimen	Specimen
Staging and scales	Staging / Scales
Standardized MedDRA Query	SMQ
Substance	Substance
System Organ Class	SOC
Therapeutic Class	Therapeutic Class
UCUM	Unit
UCUM Canonical	Canonical Unit
UCUM Custom	Unit
UCUM Standard	Unit
Undefined	Undefined
UNKNOWN	Undefined
VA Class	Drug Class
VA Drug Interaction	Drug Interaction
VA Product	Drug Product
Visit	Visit
Visit Type	Visit Type

## 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 <a href="#">CONCEPT</a> 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 <a href="#">CONCEPT</a> table associated with the relationship. Relationships are directional, and this field represents the destination concept designation.

Field	Required	Type	Description
relationship_id	Yes	varchar(20)	A unique identifier to the type or nature of the Relationship as defined in the <a href="#">RELATIONSHIP</a> 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

- Relationships can generally be classified as hierarchical (parent-child) or non-hierarchical (lateral).
- 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'.
- There is one record for each Concept Relationship connecting the same Concepts with the same relationship\_id.
- 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.
- 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 hierarchical relationships over several "generations" of direct relationships.
- In previous versions of the CDM, the relationship\_id used to be a numerical identifier. See the [RELATIONSHIP](#) table.

## 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\\_RELATIONSHIP](#) 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.

Field	Required	Type	Description
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

- There is one record for each Relationship.
- Relationships are classified as hierarchical (parent-child) or non-hierarchical (lateral)
- They are used to determine which concept relationship records should be included in the computation of the CONCEPT\_ANCESTOR table.
- The relationship\_id field contains an alphanumerical identifier, that can also be used as the abbreviation of the Relationship.
- The relationship\_name field contains the unabbreviated names of the Relationship.
- Relationships all exist symmetrically, i.e. in both direction. The relationship\_id of the opposite Relationship is provided in the reverse\_relationship\_id field.
- 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.
- 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.
- Relationships, also hierarchical, can be between Concepts within the same Vocabulary or those adopted from different Vocabulary sources.
- In past versions of the RELATIONSHIP table, the relationship\_id used to be a numerical value. A conversion table between these old and new IDs is given below:

Previous Relationship_id	Version 5 Relationship_id
1	LOINC replaced by
2	Has precise ing
3	Has tradename
4	RxNorm has dose form
5	Has form
6	RxNorm has ing
7	Constitutes
8	Contains
9	Reformulation of
10	Subsumes
11	NDFRT has dose form
12	Induces
13	May diagnose
14	Has physio effect
15	Has CI physio effect

Previous Relationship_id	Version 5 Relationship_id
16	NDFRT has ing
17	Has CI chem class
18	Has MoA
19	Has CI MoA
20	Has PK
21	May treat
22	CI to
23	May prevent
24	Has metabolites
25	Has metabolism
26	May be inhibited by
27	Has chem structure
28	NDFRT - RxNorm eq
29	Has recipient cat
30	Has proc site
31	Has priority
32	Has pathology
33	Has part of
34	Has severity
35	Has revision status
36	Has access
37	Has occurrence
38	Has method
39	Has laterality
40	Has interprets
41	Has indir morph
42	Has indir device
43	Has specimen
44	Has interpretation
45	Has intent
46	Has focus
47	Has manifestation
48	Has active ing
49	Has finding site
50	Has episodicity
51	Has dir subst
52	Has dir morph
53	Has dir device
54	Has component
55	Has causative agent
56	Has asso morph
57	Has asso finding
58	Has measurement
59	Has property
60	Has scale type
61	Has time aspect
62	Has specimen proc
63	Has specimen source
64	Has specimen morph
65	Has specimen topo
66	Has specimen subst
67	Has due to

Previous Relationship_id	Version 5 Relationship_id
68	Has relat context
69	Has dose form
70	Occurs after
71	Has asso proc
72	Has dir proc site
73	Has indir proc site
74	Has proc device
75	Has proc morph
76	Has finding context
77	Has proc context
78	Has temporal context
79	Findinga sso with
80	Has surgical appr
81	Using device
82	Using energy
83	Using subst
84	Using acc device
85	Has clinical course
86	Has route of admin
87	Using finding method
88	Using finding inform
92	ICD9P - SNOMED eq
93	CPT4 - SNOMED cat
94	CPT4 - SNOMED eq
125	MedDRA - SNOMED eq
126	Has FDA-appr ind
127	Has off-label ind
129	Has CI
130	ETC - RxNorm
131	ATC - RxNorm
132	SMQ - MedDRA
135	LOINC replaces
136	Precise ing of
137	Tradename of
138	RxNorm dose form of
139	Form of
140	RxNorm ing of
141	Consists of
142	Contained in
143	Reformulated in
144	Is a
145	NDFRT dose form of
146	Induced by
147	Diagnosed through
148	Physiol effect by
149	CI physiol effect by
150	NDFRT ing of
151	CI chem class of
152	MoA of
153	CI MoA of
154	PK of
155	May be treated by

Previous Relationship_id	Version 5 Relationship_id
156	CI by
157	May be prevented by
158	Metabolite of
159	Metabolism of
160	Inhibits effect
161	Chem structure of
162	RxNorm - NDFRT eq
163	Recipient cat of
164	Proc site of
165	Priority of
166	Pathology of
167	Part of
168	Severity of
169	Revision status of
170	Access of
171	Occurrence of
172	Method of
173	Laterality of
174	Interprets of
175	Indir morph of
176	Indir device of
177	Specimen of
178	Interpretation of
179	Intent of
180	Focus of
181	Manifestation of
182	Active ing of
183	Finding site of
184	Episodicity of
185	Dir subst of
186	Dir morph of
187	Dir device of
188	Component of
189	Causative agent of
190	Asso morph of
191	Asso finding of
192	Measurement of
193	Property of
194	Scale type of
195	Time aspect of
196	Specimen proc of
197	Specimen identity of
198	Specimen morph of
199	Specimen topo of
200	Specimen subst of
201	Due to of
202	Relat context of
203	Dose form of
204	Occurs before
205	Asso proc of
206	Dir proc site of
207	Indir proc site of



Previous Relationship_id	Version 5 Relationship_id
208	Proc device of
209	Proc morph of
210	Finding context of
211	Proc context of
212	Temporal context of
213	Asso with finding
214	Surgical appr of
215	Device used by
216	Energy used by
217	subst used by
218	Acc device used by
219	Clinical course of
220	Route of admin of
221	Finding method of
222	Finding inform of
226	SNOMED - ICD9P eq
227	SNOMED cat - CPT4
228	SNOMED - CPT4 eq
239	SNOMED - MedDRA eq
240	Is FDA-appr ind of
241	Is off-label ind of
243	Is CI of
244	RxNorm - ETC
245	RxNorm - ATC
246	MedDRA - SMQ
247	Ind/CI - SNOMED
248	SNOMED - ind/CI
275	Has therap class
276	Therap class of
277	Drug-drug inter for
278	Has drug-drug inter
279	Has pharma prep
280	Pharma prep in
281	Inferred class of
282	Has inferred class
283	SNOMED proc - HCPCS
284	HCPCS - SNOMED proc
285	RxNorm - NDFRT name
286	NDFRT - RxNorm name
287	ETC - RxNorm name
288	RxNorm - ETC name
289	ATC - RxNorm name
290	RxNorm - ATC name
291	HOI - SNOMED
292	SNOMED - HOI
293	DOI - RxNorm
294	RxNorm - DOI
295	HOI - MedDRA
296	MedDRA - HOI
297	NUCC - CMS Specialty
298	CMS Specialty - NUCC
299	DRG - MS-DRG eq

Previous Relationship_id	Version 5 Relationship_id
300	MS-DRG - DRG eq
301	DRG - MDC cat
302	MDC cat - DRG
303	Visit cat - PoS
304	PoS - Visit cat
305	VAProd - NDFRT
306	NDFRT - VAProd
307	VAProd - RxNorm eq
308	RxNorm - VAProd eq
309	RxNorm replaced by
310	RxNorm replaces
311	SNOMED replaced by
312	SNOMED replaces
313	ICD9P replaced by
314	ICD9P replaces
315	Multilex has ing
316	Multilex ing of
317	RxNorm - Multilex eq
318	Multilex - RxNorm eq
319	Multilex ing - class
320	Class - Multilex ing
321	Maps to
322	Mapped from
325	Map includes child
326	Included in map from
327	Map excludes child
328	Excluded in map from
345	UCUM replaced by
346	UCUM replaces
347	Concept replaced by
348	Concept replaces
349	Concept same_as to
350	Concept same_as from
351	Concept alt_to to
352	Concept alt_to from
353	Concept poss_eq to
354	Concept poss_eq from
355	Concept was_a to
356	Concept was_a from
357	SNOMED meas - HCPCS
358	HCPCS - SNOMED meas
359	Domain subsumes
360	Is domain

## 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.

Field	Required	Type	Description
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

- 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.
- 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.
- 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

- Each concept is also recorded as an ancestor of itself.
- 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.
- 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.
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

- 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'.
- However, this table can still be used for the translation of local source codes into Standard Concepts.
- **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).
- 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.
- 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).
- The source\_code\_description contains an optional description of the source code.
- 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
- 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.
- 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

Field	Required	Type	Description
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.
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

- The DRUG\_STRENGTH table contains information for each active (non-deprecated) standard drug concept.
- A drug which contains multiple active Ingredients will result in multiple DRUG\_STRENGTH records, one for each active ingredient.
- Ingredient strength information is provided either as absolute amount (usually for solid formulations) or as concentration (usually for liquid formulations).
- 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').
- 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').
- 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.
- 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.
- 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.

- 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 assumption is made that the density of the drug is that of water, and one can assume 1 g = 1 mL.
- All Drug vocabularies containing Standard Concepts have entries in the DRUG\_STRENGTH table.
- 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'

## 4.11 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

### 4.11.1 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.

## 4.12 ATTRIBUTE\_DEFINITION

The ATTRIBUTE\_DEFINITION table contains records defining Attributes, or covariates, to members of a Cohort through an associated description and syntax and upon instantiation (execution of the algorithm)

placed into the COHORT\_ATTRIBUTE table. Attributes are derived elements that can be selected or calculated for a subject in a Cohort. The ATTRIBUTE\_DEFINITION table provides a standardized structure for maintaining the rules governing the calculation of covariates for a subject in a Cohort, and can store operational programming code to instantiate the Attributes for a given Cohort within the OMOP Common Data Model.

Field	Required	Type	Description
attribute_definition_id	Yes	integer	A unique identifier for each Attribute.
attribute_name	Yes	varchar(255)	A short description of the Attribute.
attribute_description	No	varchar(MAX)	A complete description of the Attribute definition
attribute_type_concept_id	Yes	integer	Type defining what kind of Attribute Definition the record represents and how the syntax may be executed
attribute_syntax	No	varchar(MAX)	Syntax or code to operationalize the Attribute definition

#### 4.12.1 Conventions

- Like the definition syntax field for the COHORT\_DEFINITION table, the attribute\_definition\_syntax does not prescribe any specific syntax or programming language. Typically, it would be any flavor SQL, or a cohort definition language, or a free-text description of the algorithm.
- The Attribute Definition is generic and not necessarily related to a specific Cohort Definition, however the instantiated Attribute is linked to the Cohort records (see below the COHORT table. For example, the Attribute “Age” can be defined as the amount of time between the cohort\_start\_date of the COHORT table and the year\_of\_birth, month\_of\_birth and day\_of\_birth of the PERSON table. Thus, such a Attribute Definition can be applied and instantiated with any Cohort, as long as it is applied to a Cohort of the same Domain (Person in this case), as it is defined in the subject\_concept\_id in the COHORT\_DEFINITION table.

## 5 Standardized Metadata

### CDM\_SOURCE

All metadata about the data should be derived from the data themselves. However, the following contains a few key pieces of information that are convenient especially for software applications utilizing the CDM data.

Below provides an entity-relationship diagram highlighting the tables within the Standardized Metadata portion of the OMOP Common Data Model:



<b>cdm_source</b>	
cdm_source_name	
cdm_source_abbreviation	
cdm_holder	
source_description	
source_documentation_reference	
cdm_etl_reference	
source_release_date	
cdm_release_date	
cdm_version	
vocabulary_version	

## 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
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

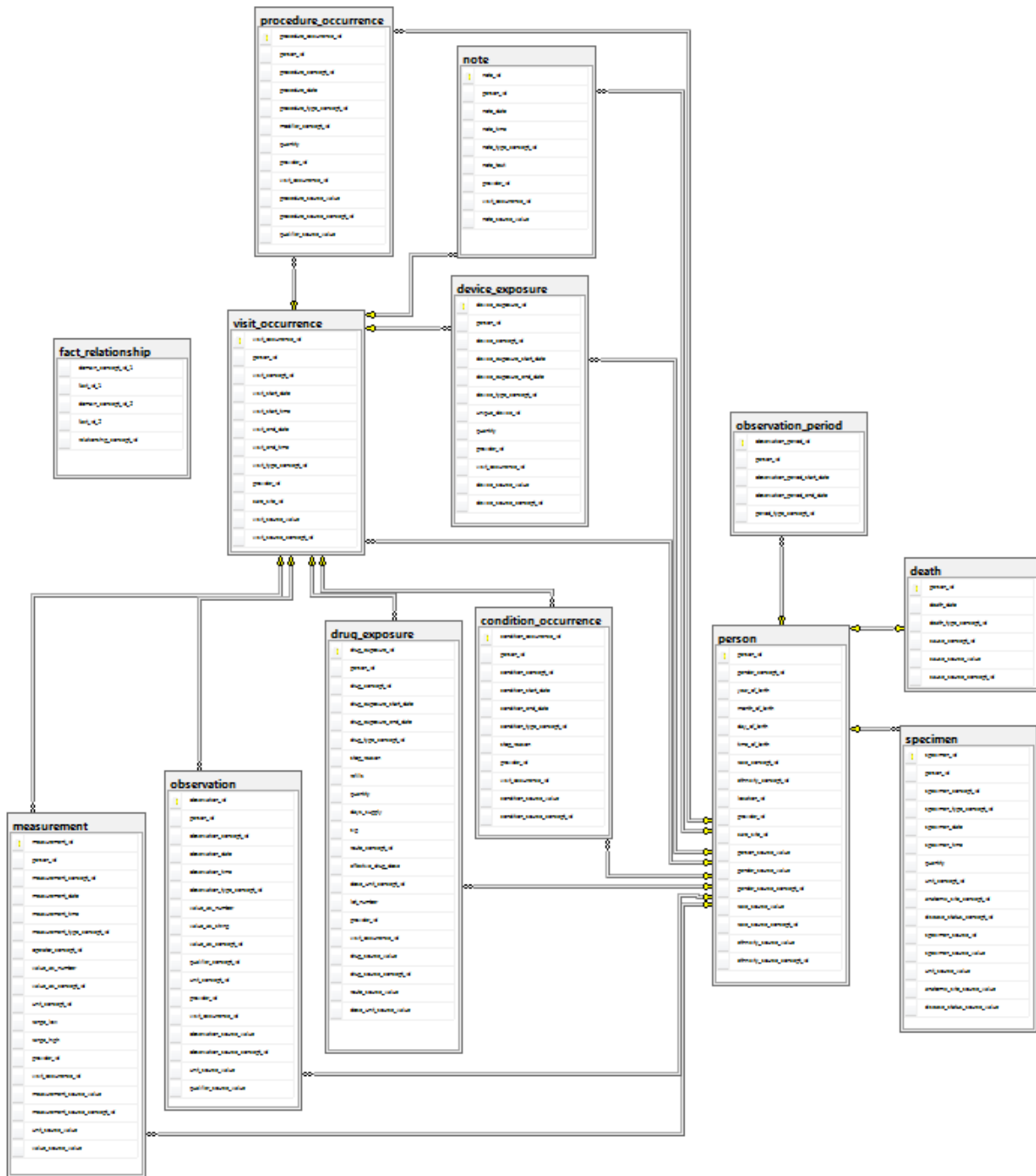
### 5.1.1 Conventions

- 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.
- Currently, there is no mechanism to link individual records in the CDM tables to their source record in the CDM\_SOURCE table.
- The version of the vocabulary can be obtained from the vocabulary\_name field in the VOCABULARY table for the record where vocabulary\_id='None'.

## 6 Standardized Clinical Data Tables

PERSON  
OBSERVATION\_PERIOD  
SPECIMEN  
DEATH  
VISIT\_OCCURRENCE  
VISIT\_DETAIL  
PROCEDURE\_OCCURRENCE  
DRUG\_EXPOSURE  
DEVICE\_EXPOSURE  
CONDITION\_OCCURRENCE  
MEASUREMENT  
NOTE  
NOTE\_NLP  
OBSERVATION  
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.

Field	Required	Type	Description
gender_concept_id	Yes	integer	A foreign key that refers to an identifier in the CONCEPT table for the unique gender of the person.
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.
race_concept_id	Yes	integer	A foreign key that refers to an identifier in the CONCEPT table for the unique race of the person.
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.
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 person'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	No	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	No	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	No	Integer	A foreign key to the ethnicity concept that refers to the code used in the source.

### 6.1.1 Conventions

- All tables representing patient-related Domains have a foreign-key reference to the person\_id field in the PERSON table.
- 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.
- Valid Gender, Race and Ethnicity Concepts each belong to their own Domain.
- Ethnicity in the OMOP CDM follows the OMB Standards for Data on Race and Ethnicity: Only distinctions between Hispanics and Non-Hispanics are made.
- Additional information is stored through references to other tables, such as the home address (location\_id) or the primary care provider.
- The Provider refers to the primary care provider (General Practitioner).
- The Care Site refers to where the Provider typically provides the primary care.

## 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.
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

### 6.2.1 Conventions

- One Person may have one or more disjoint observation periods, during which times analyses may assume that clinical events would be captured if observed, and outside of which no clinical events may be recorded.
- Each Person can have more than one valid OBSERVATION\_PERIOD record, but no two observation periods can overlap in time for a given person.
- 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.
- 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`.
- No clinical data are valid outside an active Observation Period. Clinical data that refer to a time outside (diagnoses of previous conditions such as “Old MI” or medical history) of an active Observation Period are recorded as Observations. The date of the Observation is the first day of the first Observation Period of a patient.
- For claims data, observation periods are inferred from the enrollment periods to a health benefit plan.
- 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.

## 6.3 SPECIMEN

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

Field	Required	Type	Description
<code>specimen_id</code>	Yes	integer	A unique identifier for each specimen.
<code>person_id</code>	Yes	integer	A foreign key identifier to the Person for whom the Specimen is recorded.
<code>specimen_concept_id</code>	Yes	integer	A foreign key referring to a Standard Concept identifier in the Standardized Vocabularies for the Specimen.
<code>specimen_type_concept_id</code>	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.
<code>specimen_date</code>	Yes	date	The date the specimen was obtained from the Person.
<code>specimen_datetime</code>	No	datetime	The date and time on the date when the Specimen was obtained from the person.
<code>quantity</code>	No	float	The amount of specimen collection from the person during the sampling procedure.
<code>unit_concept_id</code>	No	integer	A foreign key to a Standard Concept identifier for the Unit associated with the numeric quantity of the Specimen collection.
<code>anatomic_site_concept_id</code>	No	integer	A foreign key to a Standard Concept identifier for the anatomic location of specimen collection.
<code>disease_status_concept_id</code>	No	integer	A foreign key to a Standard Concept identifier for the Disease Status of specimen collection.
<code>specimen_source_id</code>	No	varchar(50)	The Specimen identifier as it appears in the source data.

Field	Required	Type	Description
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.

### 6.3.1 Conventions

- Anatomic site is coded at the most specific level of granularity possible, such that higher level classifications can be derived using the Standardized Vocabularies.

## 6.4 DEATH

The death domain contains the clinical event for how and when a Person dies. A person can have up to one record if the source system contains 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

Field	Required	Type	Description
person_id	Yes	integer	A foreign key identifier to the deceased person. The demographic details of that person are stored in the person table.
death_date	Yes	date	The date the person was deceased. If the precise date including day or month is not known or not allowed, December is used as the default month, and the last day of the month the default day.
death_datetime	No	datetime	The date and time the person was deceased. If the precise date including day or month is not known or not allowed, December is used as the default month, and the last day of the month the default day.
death_type_concept_id	Yes	integer	A foreign key referring to the predefined concept identifier in the Standardized Vocabularies reflecting how the death was represented in the source data.
cause_concept_id	No	integer	A foreign key referring to a standard concept identifier in the Standardized Vocabularies for conditions.
cause_source_value	No	varchar(50)	The source code for the cause of death 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.

Field	Required	Type	Description
cause_source_concept_id	No	integer	A foreign key to the concept that refers to the code used in the source. Note, this variable name is abbreviated to ensure it will be allowable across database platforms.

### 6.4.1 Conventions

- Living patients should not contain any information in the DEATH table.
- Each Person may have more than one record of death in the source data. It is the task of the ETL to pick the most plausible or most accurate records to be aggregated and stored as a single record in the DEATH table.
- If the Death Date cannot be precisely determined from the data, the best approximation should be used.
- Valid Concepts for the cause\_concept\_id have domain\_id='Condition'.

## 6.5 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.
visit_start_date	Yes	date	The start date of the visit.
visit_start_datetime	No	datetime	The date and time of the visit started.
visit_end_date	Yes	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	No	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.
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	No	integer	A foreign key to a Concept that refers to the code used in the source.
admitting_source_concept_id	No	integer	A foreign key to the predefined concept in the Place of Service Vocabulary reflecting the admitting source for a visit.



Field	Required	Type	Description
admitting_source_value	No	varchar(50)	The source code for the admitting source as it appears in the source data.
discharge_to_concept_id	No	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

### 6.5.1 Conventions

- A Visit Occurrence is recorded for each visit to a healthcare facility.
- Valid Visit Concepts belong to the “Visit” domain.
- Standard Visit Concepts are defined 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.
- Handling of death: In the case when a patient died during admission (Visit\_Occurrence. discharge\_disposition\_concept\_id = 4216643 ‘Patient died’), a record in the Death table should be created with death\_type\_concept\_id = 44818516 (EHR discharge status “Expired”).
- Source Concepts from place of service vocabularies are mapped into these standard visit Concepts in the Standardized Vocabularies.
- At any one day, there could be more than one visit.
- 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.
- 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.
- 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.
- In addition to the “Place of Service” vocabulary the following SNOMED concepts for discharge disposition can be used:
  - Patient died: 4216643
  - Absent without leave: 44814693
  - Patient self-discharge against medical advice: 4021968
- In the case where a patient died during admission (Visit\_Occurrence.discharge\_disposition\_concept\_id = 4216643 “Patient died”), a record in the Death table should be created with death\_type\_concept\_id = 44818516 (EHR discharge status “Expired”).
- PRECEDING\_VISIT\_ID can be used to link a visit immediately preceding the current visit
- Some EMR systems combine emergency room followed by inpatient admission into one visit, and it is close to impossible to separate the two. To annotate this visit type, a new visit concept “Emergency Room and Inpatient Visit” was added (CONCEPT\_ID 262).

## 6.6 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.
visit_detail_start_date	Yes	date	The start date of the visit.
visit_detail_start_datetime	No	datetime	The date and time of the visit started.
visit_detail_end_date	Yes	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	No	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.
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	No	Integer	A foreign key to a Concept that refers to the code used in the source.
admitting_source_value	No	Varchar(50)	The source code for the admitting source as it appears in the source data.
admitting_source_concept_id	No	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	No	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
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.

### 6.6.1 Conventions

- All conventions used in VISIT\_OCCURRENCE apply to VISIT\_DETAIL, some notable exceptions:
- 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.

- One record in VISIT\_DETAIL can only have one VISIT\_OCCURRENCE parent.
- A single VISIT\_OCCURRENCE record may have many child VISIT\_DETAIL records.
- 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.
- Handling of death: In the case when a patient died during admission (VISIT\_OCCURRENCE.DISCHARGE\_DISPOSITION\_ID = 4216643 ‘Patient died’), a record in the Death table should be created with DEATH\_TYPE\_CONCEPT\_ID = 44818516 (EHR discharge status “Expired”).
- Source Concepts from place of service vocabularies are mapped into these Standard Visit Concepts in the Standardized Vocabularies.
- At 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.
- 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 ETL to specify multiple child records per visit occurrence - and each of these child records may represent different provider\_ids.
- 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 ETL to specify multiple child records per visit occurrence - and each of these child records may represent different care\_sites.
- 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
- Unlike VISIT\_OCCURRENCE, VISIT\_DETAIL may have nested visits with hierarchical relationships to each other.
- Representation of US claim data: US claims data generally has two-levels. Header/summary data that summarizes the entire claim; 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 atleast one FK link from VISIT\_DETAIL to VISIT\_OCCURRENCE.

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 <math>\hat{=}</math> ER leading to ICU leading to medical floor, leading to rehabilitation, or in hierarchical parent-child visit <math>\hat{=}</math> a visit for dialysis while in ICU.

Note the CONCEPT\_ID for visit domain is 8, 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)

## 6.7 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.

Field	Required	Type	Description
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	Yes	date	The date on which the Procedure was performed.
procedure_datetime	No	datetime	The date and time on which the Procedure was performed.
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.
modifier_concept_id	No	integer	A foreign key to a Standard Concept identifier for a modifier to the Procedure (e.g. bilateral)
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	No	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.

### 6.7.1 Conventions

- 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.
- Procedures are expected to be carried out within one day and therefore have no end date.
- 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.

- If the quantity value is omitted, a single procedure is assumed.
- 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.
- The Visit during which the procedure was performed is recorded through a reference to the VISIT\_OCCURRENCE table. This information is not always available.
- 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.
- The Provider carrying out the procedure is recorded through a reference to the PROVIDER table. This information is not always available.

## 6.8 DRUG\_EXPOSURE

The drug exposure 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	integer	A system-generated unique identifier for each Drug utilization event.
person_id	Yes	integer	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 for the Drug concept.
drug_exposure_start_date	Yes	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	No	datetime	The start date and time 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_end_date	Yes	date	The end date for the current instance of Drug utilization. It is not available from all sources.

Field	Required	Type	Description
drug_exposure_end_datetime	No	datetime	The end date and time for the current instance of Drug utilization. It is not available from all sources.
verbatim_end_date	No	date	The known end date of a drug_exposure as provided by the source
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.
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 0.
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 recorded in the original prescription or dispensing record.
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	No	integer	A foreign key to a predefined concept in the Standardized Vocabularies reflecting the route of administration.
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	No	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.

### 6.8.1 Conventions

- 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.
- 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.
- 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` should contain 0.
- A Drug Type is assigned to each Drug Exposure to track from what source the information was drawn or inferred from. The valid `domain_id` for these Concepts is “Drug Type”.
- 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.
- 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. However, the `route_concept_id` could resolve ambiguities of how a certain Drug Form is actually applied. For example, a “Solution” could be used orally or parenterally, and this field will make this determination.
- The `lot_number` field contains an identifier assigned from the manufacturer of the Drug product.
- 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.
- If possible, the prescribing or administering provider (physician or nurse) is recorded in the `provider_id` field through a reference to the provider table.
- 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_date = drug_exposure_start_date + days_supply - 1`), or because the exposure was stopped (medication changed, medication discontinued, etc.)

## 6.9 DEVICE\_EXPOSURE

The device exposure domain captures information about a person’s exposure to a foreign physical object or instrument that 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
<code>device_exposure_id</code>	Yes	integer	A system-generated unique identifier for each Device Exposure.
<code>person_id</code>	Yes	integer	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.

Field	Required	Type	Description
device_concept_id	Yes	integer	A foreign key that refers to a Standard Concept identifier in the Standardized Vocabularies for the Device concept.
device_exposure_start_date	Yes	date	The date the Device or supply was applied or used.
device_exposure_start_datetime	No	datetime	The date and time the Device or supply was applied or used.
device_exposure_end_date	No	date	The date the Device or supply was removed from use.
device_exposure_end_datetime	No	datetime	The date and time the Device or supply was removed from use.
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.
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 for 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 in the VISIT_DETAIL table during which the Drug Exposure was initiated.
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	No	integer	A foreign key to a Device Concept that refers to the code used in the source.

### 6.9.1 Conventions

- 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.
- For medical devices that are regulated by the FDA, if a Unique Device Identification (UDI) is provided if available in the data source, and is recorded in the `unique_device_id` field.
- 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.
- A Device Type is assigned to each Device Exposure to track from what source the information was drawn or inferred. The valid `domain_id` for these Concepts is “Device Type”.
- The Visit during which the Device was first used is recorded through a reference to the `VISIT_OCCURRENCE` table. This information is not always available.
- The Visit Detail during which the Device was first used is recorded through a reference to the



VISIT\_DETAIL table. This information is not always available.

- The Provider exposing the patient to the Device is recorded through a reference to the PROVIDER table. This information is not always available.

## 6.10 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 ICD-9-CM that are submitted as part of a reimbursement claim for health services and
- EHRs may capture Person Conditions in the form of diagnosis codes or symptoms.

Field	Required	Type	Description
condition_occurrence_id	Yes	integer	A unique identifier for each Condition Occurrence event.
person_id	Yes	integer	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 Condition Concept identifier in the Standardized Vocabularies.
condition_start_date	Yes	date	The date when the instance of the Condition is recorded.
condition_start_datetime	No	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	date	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.
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.

Field	Required	Type	Description
condition_source_concept_id	No	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.
condition_status_concept_id	No	integer	A foreign key to the predefined Concept in the Standard Vocabulary reflecting the condition status

### 6.10.1 Conventions

- Valid Condition Concepts belong to the “Condition” domain.
- Condition records are typically inferred from diagnostic codes recorded in the source data. Such code system, 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 instead.
- 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.
- Family history and past diagnoses (“history of”) are not recorded in the `CONDITION_OCCURRENCE` table. Instead, they are listed in the `OBSERVATION` table.
- 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.
- 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.
- 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.
- Condition source codes are typically ICD-9-CM, Read or ICD-10 diagnosis codes from medical claims or discharge status/visit diagnosis codes from EHRs.
- 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

## 6.11 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.
measurement_date	Yes	date	The date of the Measurement.
measurement_datetime	No	datetime	The date and time of the Measurement. Some database systems don't have a datatype of time. To accomodate all temporal analyses, datatype datetime can be used (combining measurement_date and measurement_time <a href="#">forum discussion</a> )
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.
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 <, <=, =, >=, >.
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.).
unit_concept_id	No	integer	A foreign key to a Standard Concept ID of Measurement Units in the Standardized Vocabularies.
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.
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.

Field	Required	Type	Description
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	No	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.

### 6.11.1 Conventions

- 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.
- 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.
- 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.
- 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).
- Valid Concepts for the value\_as\_concept field belong to the ‘Meas Value’ domain.
- 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”.
- 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.
- 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 =.

- Valid Concepts for the operator\_concept\_id field belong to the ‘Meas Value Operator’ domain.
- The Unit is optional even if a value\_as\_number is provided.
- 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.
- The Visit during which the observation was made is recorded through a reference to the VISIT\_OCCURRENCE table. This information is not always available.
- The Provider making the observation is recorded through a reference to the PROVIDER table. This information is not always available.

## 6.12 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_date	Yes	date	The date the note was recorded.
note_datetime	No	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.
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.
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

### 6.12.1 Conventions

- The NOTE table contains free text (in ASCII, or preferably in UTF8 format) taken by a healthcare Provider.
- The Visit during which the note was written is recorded through a reference to the VISIT\_OCCURRENCE

table. This information is not always available.

- The Provider making the note is recorded through a reference to the PROVIDER table. This information is not always available.
- The type of note\_text is CLOB or varchar(MAX) depending on RDBMS
- 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.

### 6.12.2 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 CMSi's 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 should roll up to a unique LOINC code. For example, Dentistry Hygienist Outpatient Progress note (LOINC code 34127-1) has the following dimensions:

- 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.<sup>2</sup> HL7/LOINC workforce is committed to establish new LOINC codes for each new encountered combination of CDO dimensions.<sup>3</sup>

Automation of mapping of clinical notes to a standard terminology based on the note title is possible when it is driven by ontology (aka CDO). Mapping to individual LOINC codes which may or may not exist for a particular note type cannot be fully automated. To support mapping of clinical notes to CDO in OMOP CDM, we propose the following approach:

#### 6.12.2.1 1. Add all LOINC concepts representing 5 CDO dimensions to the Concept table. For example:

Field	Record 1	Record 2
concept_id	55443322132	55443322175
concept_name	Administrative note	Against medical advice note
concept_code	LP173418-7	LP173388-2
vocabulary_id	LOINC	LOINC

#### 6.12.2.2 2. Represent CDO hierarchy in the Concept\_Relationship table using the Subsumes relationship pair. For example:

Field	Record 1	Record 2
concept_id_1	55443322132	55443322175

Field	Record 1	Record 2
concept_id_2	55443322175	55443322132
relationship_id	Subsumes	Is a

**6.12.2.3 3. Add LOINC document codes to the Concept table (e.g. Dentistry Hygienist Outpatient Progress note, LOINC code 34127-1). For example:**

Field	Record 1	Record 2
concept_id	193240	193241
concept_name	Dentistry Hygienist Outpatient Progress note	Consult note
concept_code	34127-1	11488-4
vocabulary_id	LOINC	LOINC

**6.12.2.4 4. Represent dimensions of each document concept in Concept\_Relationship table by its relationships to the respective concepts from CDO.**

- Use the `Member Of` (new) relationship pair.
- Using example from the Dentistry Hygienist Outpatient Progress note (LOINC code 34127-1):

concept_id_1	concept_id_2	relationship_id
193240	55443322132	Member Of
55443322132	193240	Has Member
193240	55443322175	Member Of
55443322175	193240	Has Member
193240	55443322166	Member Of
55443322166	193240	Has Member
193240	55443322107	Member Of
55443322107	193240	Has Member
193240	55443322146	Member Of
55443322146	193240	Has Member

Where concept codes represent the following concepts:

Content	Description
193240	Corresponds to LOINC 34127-1, Dentistry Hygienist Outpatient Progress note
55443322132	Corresponds to LOINC LP173418-7, Kind of Document = Note
55443322175	Corresponds to LOINC LP173213-2, Type of Service = Progress
55443322166	Corresponds to LOINC LP173051-6, Setting = Outpatient
55443322107	Corresponds to LOINC LP172934-4, Subject Matter Domain = Dentistry
55443322146	Corresponds to LOINC LP173071-4, Role = Hygienist

Most of the codes will not have all 5 dimensions. Therefore, they may be represented by 2-5 relationship pairs.

**6.12.2.5 5. If LOINC does not have a code corresponding to a permutation of the 5 CDO encountered in the source, this code will be generated as OMOP vocabulary code.**

- Its relationships to the CDO dimensions will be represented exactly as those of existing LOINC concepts

(as described above). If/when a proper LOINC code for this permutation is released, the old code should be deprecated. Transition between the old and new codes should be represented by `i½Concept` replaced by `i½` pairs.

### 6.12.2.6 6. Mapping from the source data will be performed to the 2-5 CDO dimensions.

Query below finds LOINC code for Dentistry Hygienist Outpatient Progress note (see example above) that has all 5 dimensions:

```
SELECT
FROM Concept_Relationship
WHERE relationship_id = i½Has Memberi½ AND
      (concept_id_1 = 55443322132
      OR concept_id_1 = 55443322175
      OR concept_id_1 = 55443322166
      OR concept_id_1 = 55443322107
      OR concept_id_1 = 55443322146)
GROUP BY concept_ID_2
```

If less than 5 dimensions are available, `HAVING COUNT(n)` clause should be added to get a unique record at the intersection of these dimensions. `n` is the number of dimensions available:

```
SELECT
FROM Concept_Relationship
WHERE relationship_id = i½Has Memberi½ AND
      (concept_id_1 = 55443322132
      OR concept_id_1 = 55443322175
      OR concept_id_1 = 55443322146)
GROUP BY concept_ID_2
HAVING COUNT(*) = 3
```

## 6.13 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
<code>note_nlp_id</code>	Yes	integer	A unique identifier for each term extracted from a note.
<code>note_id</code>	Yes	integer	A foreign key to the Note table note the term was extracted from.
<code>section_concept_id</code>	No	integer	A foreign key to the predefined Concept in the Standardized Vocabularies representing the section of the extracted term.
<code>snippet</code>	No	<code>varchar(250)</code>	A small window of text surrounding the term.
<code>offset</code>	No	<code>varchar(50)</code>	Character offset of the extracted term in the input note.
<code>lexical_variant</code>	Yes	<code>varchar(250)</code>	Raw text extracted from the NLP tool.
<code>note_nlp_concept_id</code>	No	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.



Field	Required	Type	Description
note_nlp_source_concept_id	No	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.
term_temporal	No	varchar(50)	An optional time modifier associated with the extracted term. (for now “past” or “present” 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. “son has rash” ? “negated=no,subject=family, certainty=undef,conditional=false,general=false”).

### 6.13.1 Conventions

**Term\_exists** 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
- 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)

**Term\_temporal** Term\_temporal is to indicate if a condition is “present” or just in the “past”.

The following would be past:

- History = true
- Concept\_date = anything before the time of the report

**Term\_modifiers** 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.

## 6.14 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	Yes	date	The date of the observation.
observation_datetime	No	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.
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.

Field	Required	Type	Description
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	No	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

### 6.14.1 Conventions

- 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.
- 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.
- Observation 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) or a verbatim string (value\_as\_string). 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.
- 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.
- Valid Concepts of the value\_as\_concept field are not enforced, but typically belong to the “Meas Value” domain.
- For numerical facts a Unit can be provided in the unit\_concept\_id.
- For facts represented as Concepts no domain membership is enforced.
- 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”.
- The qualifier\_concept\_id field contains all attributes specifying the clinical fact further, such as as degrees, severities, drug-drug interaction alerts etc.

- The Visit during which the observation was made is recorded through a reference to the VISIT\_OCCURRENCE table. This information is not always available.
- The Provider making the observation is recorded through a reference to the PROVIDER table. This information is not always available.

## 6.15 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 (table), 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.

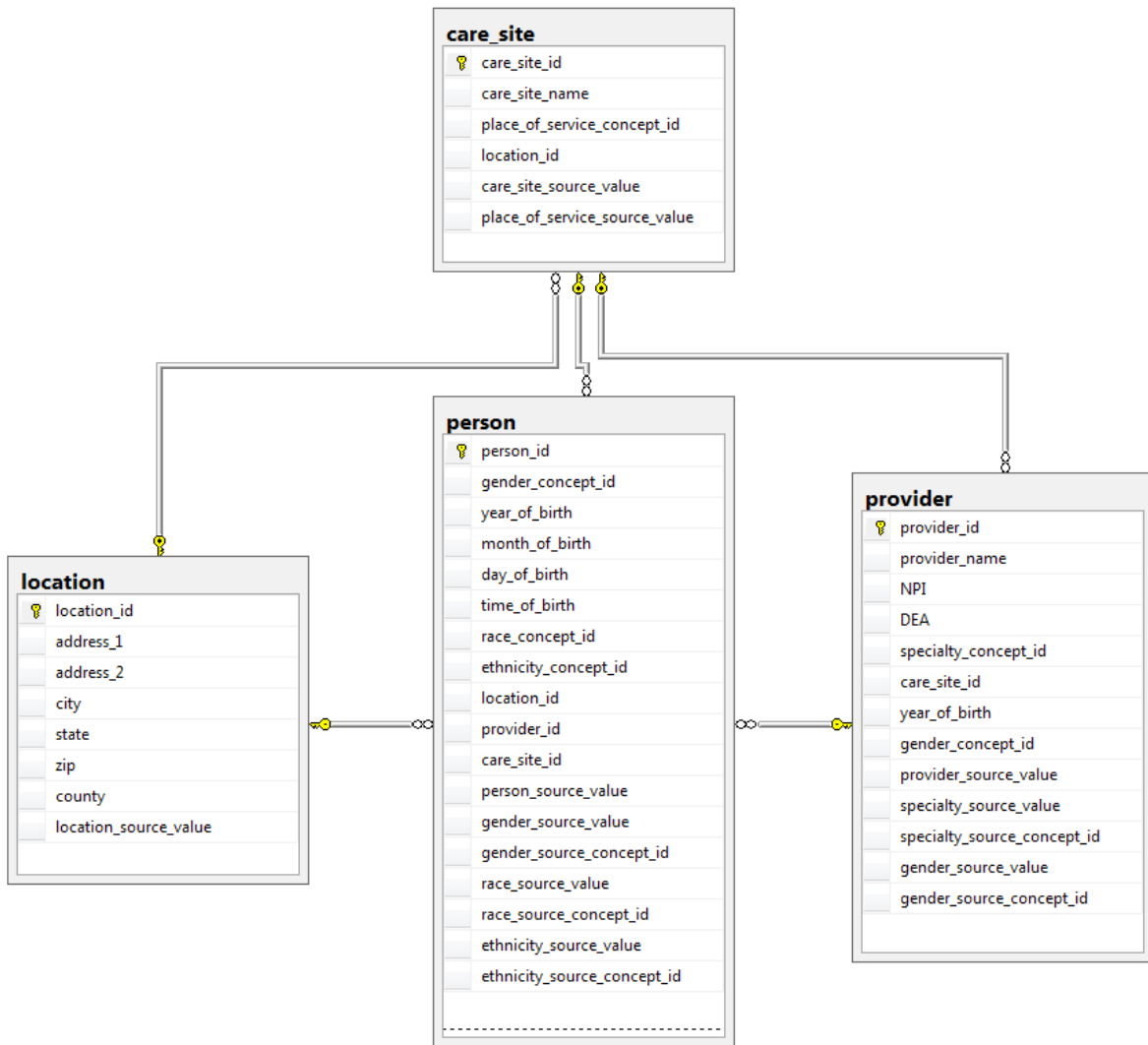
### 6.15.1 Conventions

- 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

## 7 Standardized Health System Data Tables

LOCATION  
 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:



## 7.1 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.
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.

Field	Required	Type	Description
zip	No	varchar(9)	The zip or postal code.
county	No	varchar(20)	The county.
location_source_value	No	varchar(50)	The verbatim information that is used to uniquely identify the location as it appears in the source data.

### 7.1.1 Conventions

- Each address or Location is unique and is present only once in the table.
- 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.
- 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.
- 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.
- 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.
- No country information is expected as source data are always collected within a single country.

## 7.2 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
place_of_service_concept_id	No	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.

Field	Required	Type	Description
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.

### 7.2.1 Conventions

- Care site is a unique combination of location\_id and place\_of\_service\_source\_value.
- Every record in the visit\_occurrence table may have only one care site
- Care site does not take into account the provider (human) information such a specialty.
- Many source data do not make a distinction between individual and institutional providers. The CARE\_SITE table contains the institutional providers.
- 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.
- 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.
- The relationships between Care Sites are defined in the FACT\_RELATIONSHIP table.
- The Care Site Source Value typically contains the name of the Care Site.
- The Place of Service Concepts belongs to the Domain ‘Place of Service’.

## 7.3 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	No	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	No	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	No	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.

Field	Required	Type	Description
gender_source_concept_id	No	integer	A foreign key to a Concept that refers to the code used in the source.

### 7.3.1 Conventions

- Many sources do not make a distinction between individual and institutional providers. The PROVIDER table contains the individual providers.
- 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.
- 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.
- Valid Specialty Concepts belong to the ‘Specialty’ domain.
- 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.

## 8 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.

### 8.1 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.
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	No	integer	A foreign key that refers to a standard Payer concept identifier in the Standardized Vocabularies



Field	Required	Type	Description
payer_source_value	No	varchar(50)	The source code for the payer as it appears in the source data.
payer_source_concept_id	No	integer	A foreign key to a payer concept that refers to the code used in the source.
plan_concept_id	No	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	No	integer	A foreign key to a plan concept that refers to the plan code used in the source data.
sponsor_concept_id	No	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	No	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	No	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	No	integer	A foreign key to a stop-coverage concept that refers to the code used in the source.

### 8.1.1 Conventions

- 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.
- 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.

## 8.2 COST

The COST table captures records containing the cost of any medical entity recorded in one of the DRUG\_EXPOSURE, PROCEDURE\_OCCURRENCE, VISIT\_OCCURRENCE or DEVICE\_OCCURRENCE tables. It replaces the corresponding DRUG\_COST, PROCEDURE\_COST, VISIT\_COST or DEVICE\_COST tables that were initially defined for the OMOP CDM V5. However, it also allows to capture cost information for records of the OBSERVATION and MEASUREMENT tables.

The information about the cost is defined by the amount of money paid by the Person and Payer, or as the charged cost by the healthcare provider. So, the COST table can be used to represent both cost and revenue perspectives. The `cost_type_concept_id` field will use concepts in the Standardized Vocabularies to designate the source of the cost data. A reference to the health plan information in the `PAYER_PLAN_PERIOD` table is stored in the record that is responsible for the determination of the cost as well as some of the payments.

Field	Required	Type	Description
<code>cost_id</code>	Yes	integer	A unique identifier for each COST record.
<code>cost_event_id</code>	Yes	integer	A foreign key identifier to the event (e.g. Measurement, Procedure, Visit, Drug Exposure, etc) record for which cost data are recorded.
<code>cost_domain_id</code>	Yes	varchar(20)	The concept representing the domain of the cost event, from which the corresponding table can be inferred that contains the entity for which cost information is recorded.
<code>cost_type_concept_id</code>	Yes	integer	A foreign key identifier to a concept in the <code>CONCEPT</code> table for the provenance or the source of the COST data: Calculated from insurance claim information, provider revenue, calculated from cost-to-charge ratio, reported from accounting database, etc.
<code>currency_concept_id</code>	No	integer	A foreign key identifier to the concept representing the 3-letter code used to delineate international currencies, such as USD for US Dollar.
<code>total_charge</code>	No	float	The total amount charged by some provider of goods or services (e.g. hospital, physician pharmacy, dme provider) to payers (insurance companies, the patient).
<code>total_cost</code>	No	float	The cost incurred by the provider of goods or services.
<code>total_paid</code>	No	float	The total amount actually paid from all payers for goods or services of the provider.
<code>paid_by_payer</code>	No	float	The amount paid by the Payer for the goods or services.
<code>paid_by_patient</code>	No	float	The total amount paid by the Person as a share of the expenses.
<code>paid_patient_copay</code>	No	float	The amount paid by the Person as a fixed contribution to the expenses.
<code>paid_patient_coinsurance</code>	No	float	The amount paid by the Person as a joint assumption of risk. Typically, this is a percentage of the expenses defined by the Payer Plan after the Person's deductible is exceeded.
<code>paid_patient_deductible</code>	No	float	The amount paid by the Person that is counted toward the deductible defined by the Payer Plan. <code>paid_patient_deductible</code> does contribute to the <code>paid_by_patient</code> variable.
<code>paid_by_primary</code>	No	float	The amount paid by a primary Payer through the coordination of benefits.

Field	Required	Type	Description
paid_ingredient_cost	No	float	The amount paid by the Payer to a pharmacy for the drug, excluding the amount paid for dispensing the drug. paid_ingredient_cost contributes to the paid_by_payer field if this field is populated with a nonzero value.
paid_dispensing_fee	No	float	The amount paid by the Payer to a pharmacy for dispensing a drug, excluding the amount paid for the drug ingredient. paid_dispensing_fee contributes to the paid_by_payer field if this field is populated with a nonzero value.
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.
amount_allowed	No	float	The contracted amount agreed between the payer and provider.
revenue_code_concept_id	No	integer	A foreign key referring to a Standard Concept ID in the Standardized Vocabularies for Revenue codes.
revenue_code_source_value	No	varchar(50)	The source code for the Revenue code as it appears in the source data, stored here for reference.
drg_concept_id	No	integer	A foreign key to the predefined concept in the DRG Vocabulary reflecting the DRG for a visit.
drg_source_value	No	varchar(3)	The 3-digit DRG source code as it appears in the source data.

### 8.2.1 Conventions

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) paid or reimbursed amounts, 2) charges or list prices (such as Average Wholesale Prices), and 3) costs or expenses incurred by the provider. The defined fields are variables found in almost all U.S.-based claims data sources, which is the most common data source for researchers. Non-U.S.-based data holders are encouraged to engage with OHDSI to adjust these tables to their needs.

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. Goods or services services not covered by a payer are indicated by 0 values in the amount\_allowed and patient responsibility fields (copay, coinsurance, deductible) as well as a missing payer\_plan\_period\_id. This means the patient is responsible for the total\_charged value.

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\_domain\_id field:

cost_domain_id	corresponding CDM table
Drug	DRUG_EXPOSURE

cost_domain_id	corresponding CDM table
Visit	VISIT_OCCURRENCE
Procedure	PROCEDURE_OCCURRENCE
Device	DEVICE_EXPOSURE
Measurement	MEASUREMENT
Observation	OBSERVATION
Specimen	SPECIMEN

- `cost_type_concept_id`: The concept referenced in this field defines the source of the cost information, and therefore the perspective. It could be from the perspective of the payer, or the perspective of the provider. Therefore, “cost” really means either cost or revenue, and the direction of funds (incoming and outgoing) as well as the modus of its calculation is defined by this field.
- `total_charged` and `total_cost`: 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](#) [here](#).
- `total_paid`: This field is calculated using the following formula: `paid_by_payer + paid_by_patient + paid_by_primary`. In claims data, this field is considered the calculated field the payer expects the provider to get reimbursed for goods and services, based on the payer’s contractual obligations.
- 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). The latter is usually very small and typically not provided by most source data, and therefore not included in the CDM.
- `paid_by_payer`: 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.
- `paid_by_patient`: This field is most often used in claims data to report the contracted amount the patient is responsible for reimbursing the provider for the goods and services she received. This is a calculated field using the following formula: `paid_patient_copay + paid_patient_coinsurance + paid_patient_deductible`. If the source data has actual patient payments then the patient payment should have its own cost record with a `payer_plan_id` set to 0 to indicate the the payer is actually the patient, and the actual patient payment should be noted under the `total_paid` field. The `paid_by_patient` field is only used for reporting a patient’s responsibility reported on an insurance claim.
- `paid_patient_copay` does contribute to the `paid_by_patient` variable. The `paid_patient_copay` field is only used for reporting a patient’s copay amount reported on an insurance claim.
- `paid_patient_coinsurance` does contribute to the `paid_by_patient` variable. The `paid_patient_coinsurance` field is only used for reporting a patient’s coinsurance amount reported on an insurance claim.
- `paid_patient_deductible` does contribute to the `paid_by_patient` variable. The `paid_patient_deductible` field is only used for reporting a patient’s deductible amount reported on an insurance claim.
- `amount_allowed`: This information is generally available in claims data. This is similar to the `total_paid` amount in that it shows what the payer expects the provider to be reimbursed after the payer and patient pay. This differs from the `total_paid` amount in that it is not a calculated field, but a field available directly in claims data. The field is payer-specific and the payer should be indicated by the `payer_plan_id` field.
- `paid_by_primary` does contribute to the `total_paid` variable. The `paid_by_primary` field is only used for reporting a patient’s primary insurance payment amount reported on the secondary payer insurance claim. If the source data has actual primary insurance payments (e.g. the primary insurance payment is not a derivative of the payer claim and there is verification another insurance company paid an amount to the provider), then the primary insurance payment should have its own cost record with a `payer_plan_id` set to the applicable payer, and the actual primary insurance payment should be noted under the `paid_by_payer` field.

- `revenue_code_concept_id`: Revenue codes are a method to charge for a class of procedures and conditions in the U.S. hospital system.
- `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.

## 9 Standardized Derived Elements

`COHORT`  
`COHORT_ATTRIBUTE`  
`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:



## 9.1 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.

### 9.1.1 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

## 9.2 COHORT\_ATTRIBUTE

The COHORT\_ATTRIBUTE table contains attributes associated with each subject within a cohort, as defined by a given set of criteria for a duration of time. The definition of the Cohort Attribute is contained in the ATTRIBUTE\_DEFINITION table.

Field	Required	Type	Description
cohort_definition_id	Yes	integer	A foreign key to a record in the <a href="#">COHORT_DEFINITION</a> 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.

Field	Required	Type	Description
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.
attribute_definition_id	Yes	integer	A foreign key to a record in the <a href="#">ATTRIBUTE_DEFINITION</a> table containing relevant Attribute Definition information.
value_as_number	No	float	The attribute result stored as a number. This is applicable to attributes where the result is expressed as a numeric value.
value_as_concept_id	No	integer	The attribute result stored as a Concept ID. This is applicable to attributes where the result is expressed as a categorical value.

### 9.2.1 Conventions

- Each record in the COHORT\_ATTRIBUTE table is linked to a specific record in the COHORT table, identified by matching cohort\_definition\_id, subject\_id, cohort\_start\_date and cohort\_end\_date fields.
- It adds to the Cohort records calculated co-variates (for example age, BMI) or composite scales (for example Charleson index).
- The unifying definition or feature of the Cohort Attribute is captured in the attribute\_definition\_id referring to a record in the ATTRIBUTE\_DEFINITION table.
- The actual result or value of the Cohort Attribute (co-variate, index value) is captured in the value\_as\_number (if the value is numeric) or the value\_as\_concept\_id (if the value is a concept) fields.

## 9.3 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.

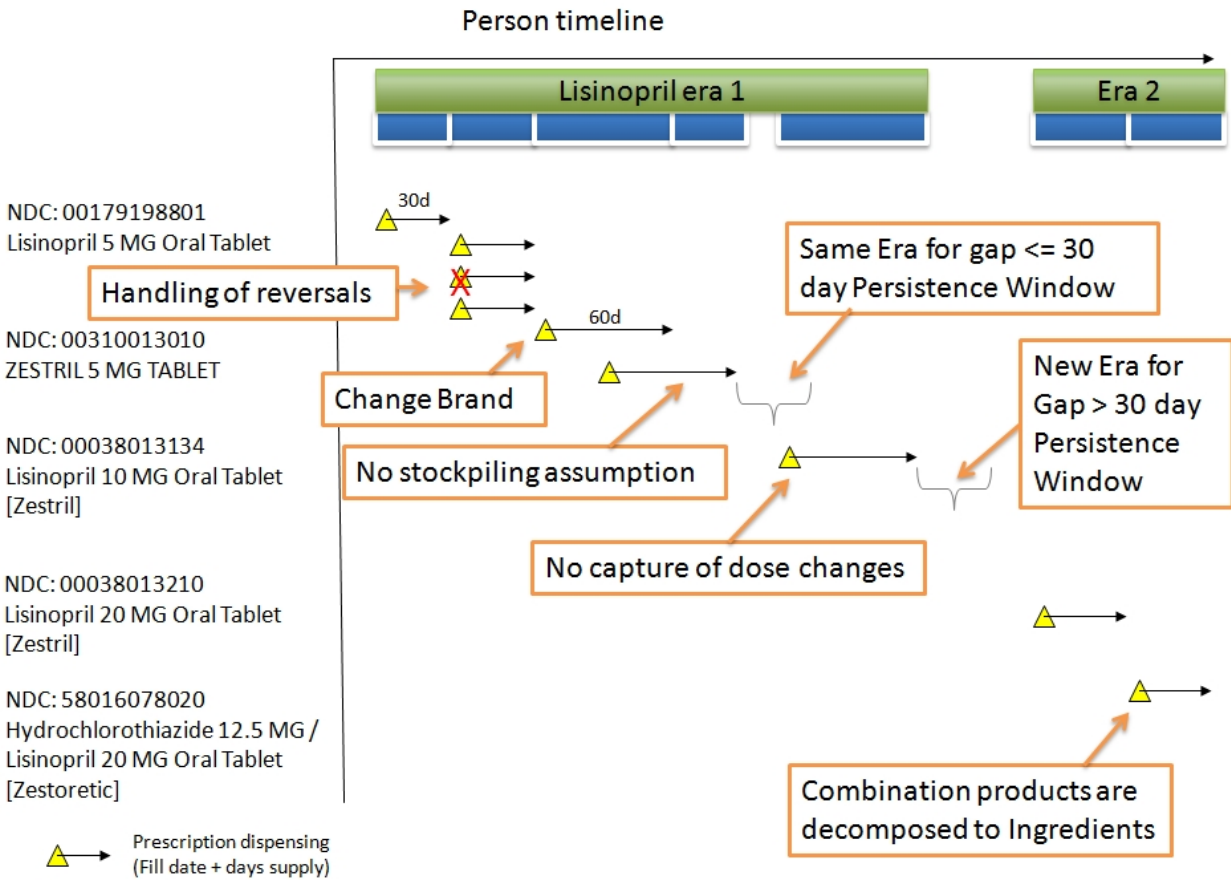


Field	Required	Type	Description
drug_era_start_date	Yes	date	The start date for the Drug Era constructed from the individual instances of Drug Exposures. It is the start date of the very first chronologically recorded instance of utilization of a Drug.
drug_era_end_date	Yes	date	The end date for the 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.

### 9.3.1 Conventions

- Drug Eras are derived from records in the DRUG\_EXPOSURE table using a standardized algorithm.
- Each Drug Era corresponds to one or many Drug Exposures that form a continuous interval and contain the same Drug Ingredient (active compound).
- 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.
- The Drug Era Start Date is the start date of the first Drug Exposure.
- 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.
- 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.
- 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.
- 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, may 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.



## 9.4 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.

Field	Required	Type	Description
dose_value	Yes	float	The numeric value of the dose.
dose_era_start_date	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_date	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.

### 9.4.1 Conventions

- 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.
- 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.
- 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.
- 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:

1	Tablets and other fixed amount formulations
	<i>Example: Acetaminophen (Paracetamol) 500 mg, 20 tablets.</i>
DRUG_STRENGTH	The denominator_unit is empty
DRUG_EXPOSURE	The quantity refers to number of pieces, e.g. tablets <i>In the example: 20</i>
Ingredient dose=	$quantity \times amount\_value [amount\_unit\_concept\_id]$ <i>Acetaminophen dose = 20 x 500mg = 10,000mg</i>
2	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=	$quantity \times numerator\_value [numerator\_unit\_concept\_id]$

3	Quantified Drugs which are formulated as a concentration
	<i>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.</i>
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. <i>Example: 2</i>
Ingredient dose=	$\text{quantity} \times \text{numerator\_value}$ [numerator_unit_concept_id] <i>Acetaminophen dose = 2 x 1250mg = 2500mg</i>
4	Drugs with the total amount provided in quantity, e.g. chemotherapeutics
	<i>Example: 42799258 "Benzyl Alcohol 0.1 ML/ML / Pramoxine hydrochloride 0.01 MG/MG Topical Gel" dispensed in a 1.25oz pack.</i>
DRUG_STRENGTH	The denominator_unit is either mg or mL. <i>Example: Benzyl Alcohol in mL and Pramoxine hydrochloride in mg</i>
DRUG_EXPOSURE	The quantity refers to mL or g. <i>Example: 1.25 x 30 (conversion factor oz -&gt; mL) = 37</i>
Ingredient dose=	$\text{quantity} \times \text{numerator\_value}$ [numerator_unit_concept_id] <i>Benzyl Alcohol dose = 37 x 0.1mL = 3.7mL</i> <i>Pramoxine hydrochloride dose = 37 x 0.01mg x 1000 = 370mg</i> <i>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.</i>
5	Compounded drugs
	<i>Example: Ibuprofen 20%/Piroxicam 1% Cream, 30ml in 5ml tubes.</i>
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. <i>Example: 20% Ibuprofen of 30ml = 6mL, 1% Piroxicam of 30ml = 0.3mL</i>
Ingredient dose=	Depends on the drugs involved: One of the use cases above. <i>Ibuprofen dose = 6 x 1mg x 1000 = 6000mg</i> <i>Piroxicam dose = 0.3 x 1mg x 1000 = 300mg</i> <i>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.</i>

6	Drugs with the active ingredient released over time, e.g. patches
DRUG_STRENGTH	<i>Example: Ethinyl Estradiol 0.000833 MG/HR / norelgestromin 0.00625 MG/HR Weekly Transdermal Patch</i> The denominator units refer to hour. <i>Example: Ethinyl Estradiol 0.000833 mg/h / norelgestromin 0.00625 mg/h</i>
DRUG_EXPOSURE	The quantity refers to the number of pieces. <i>Example: 1 patch</i>
Ingredient rate=	numerator_value [numerator_unit_concept_id] <i>Ethinyl Estradiol rate = 0.000833 mg/h norelgestromin rate 0.00625 mg/h</i> <i>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)</i>

## 9.5 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_date	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.

Field	Required	Type	Description
condition_era_end_date	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.

### 9.5.1 Conventions

- Condition Era records will be derived from the records in the `CONDITION_OCCURRENCE` table using a standardized algorithm.
- 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.
- 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`.