OMOP CDM v5.4

This is the specification document for the OMOP Common Data Model, v5.4. **This is the latest version of the OMOP CDM.** Each table is represented with a high-level description and ETL conventions that should be followed. This is continued with a discussion of each field in each table, any conventions related to the field, and constraints that should be followed (like primary key, foreign key, etc). All tables should be instantiated in a CDM instance but do not need to be populated. Similarly, fields that are not required should exist in the CDM table but do not need to be populated. Should you have questions please feel free to visit the forums or the github issue page.

Current Support for CDM v5.4

The table below details which OHDSI tools support CDM v5.4. There are two levels of support: legacy support means that the tool supports all tables and fields that were present in CDM v5.3 and feature support indicates that the tool supports any new tables and fields in CDM v5.4 that were not present in CDM v5.3. A green check ✔️ indicates that the support level for the listed tool is in place, has been tested, and released. A warning sign ❗ indicates that the support level for the listed tool has been initiated but has not yet been tested and released.

Tool	Description	Legacy Support	Feature Support
CDM R package	This package can be downloaded from https://github.com/OHDSI/CommonDataModel/. It functions to dynamically create the OMOP CDM documentation and DDL scripts to instantiate the CDM tables.	✔️	✔️

Data Quality Dashboard	This package can be downloaded from https://github.com/OHDSI/DataQualityDashboard. It runs a set of > 3500 data quality checks against an OMOP CDM instance and is required to be run on all databases prior to participating in an OHDSI network research study.	✔️	❗
Achilles	This package can be downloaded from https://github.com/OHDSI/Achilles, performing a set of broad database characterizations against an OMOP CDM instance.	✔️	❗
ARES	This package can be downloaded from https://github.com/OHDSI/Ares and is designed to display the results from both the ACHILLES and DataQualityDashboard packages to support data quality and characterization research.	✔️	❗
ATLAS	ATLAS is an open source software tool for researchers to conduct scientific analyses on standardized observational data. Demo	✔️	❗
Rabbit-In-A-Hat	This package can be downloaded from https://github.com/OHDSI/WhiteRabbit and is an application for interactive design of an ETL to the OMOP Common Data Model with the help of the the scan report generated by White Rabbit.	✔️	✔️
Feature Extraction	This package can be downloaded from https://github.com/OHDSI/FeatureExtraction. It is	✔️	✔️*

	designed to generate features (covariates) for a cohort generated using the OMOP CDM.		
Cohort Diagnostics	This package can be downloaded from https://github.com/OHDSI/CohortDiagnostics and is used to critically evaluate cohort phenotypes.	✔	❗

* The **Feature Extraction** package supports all relevant new features in CDM v5.4. For example, it was decided that, from a methodological perspective, the EPISODE and EPISODE_EVENT tables should not be included to define cohort covariates because the events that make up episodes are already pulled in as potential covariates.

Looking to send us a pull request for a bug fix? Please see the readme on the main github page.

person

Table Description

This table serves as the central identity management for all Persons in the database. It contains records that uniquely identify each person or patient, and some demographic information.

User Guide

All records in this table are independent Persons.

ETL Conventions

All Persons in a database needs one record in this table, unless they fail data quality requirements specified in the ETL. Persons with no Events should have a record nonetheless. If more than one data source contributes Events to the database, Persons must be reconciled, if possible, across the sources to create one single record per Person. The content of the BIRTH_DATETIME must be equivalent to the content of BIRTH_DAY, BIRTH_MONTH and BIRTH_YEAR.

For detailed conventions for how to populate this table, please refer to the THEMIS repository.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
person_id	It is assumed that every person with a different unique identifier is in fact a different person and should be treated independently.	Any person linkage that needs to occur to uniquely identify Persons ought to be done prior to writing this table. This identifier can be the original id from the source data provided if it is an integer, otherwise it can be an autogenerated number.	integer	Yes	Yes	No		
gender_concept_id	This field is meant to capture the biological sex at birth of the Person. This field should not be used to study gender	Use the gender or sex value present in the data under the assumption that it is the biological sex at birth. If the source data captures gender identity it should be stored in the OBSERVATION table. Accepted gender concepts. Please refer to the THEMIS repository for detailed	integer	Yes	No	Yes	CONCEPT	Gender

	identity issues.	conventions on how to populate this field.				
year_of_birth	Compute age using year_of_birth.	For data sources with date of birth, the year should be extracted. If no year of birth is available all the person’s data should be dropped from the CDM instance. For additional information on how to populate this field, please refer to the THEMIS repository.	integer	Yes	No	No
month_of_birth		For data sources that provide the precise date of birth, the month should be extracted and stored in this field.	integer	No	No	No
day_of_birth		For data sources that provide the precise date of birth, the day should be extracted and stored in this field.	integer	No	No	No
birth_datetime		This field is not required but highly encouraged. For data sources that provide the precise datetime of birth, that value should be stored in this field. For more information	datetime	No	No	No

		on how to populate this field, please refer to the THEMIS repository.						
race_concept_id	This field captures race or ethnic background of the person.	Only use this field if you have information about race or ethnic background. The Vocabulary contains Concepts about the main races and ethnic backgrounds in a hierarchical system. Due to the imprecise nature of human races and ethnic backgrounds, this is not a perfect system. Mixed races are not supported. If a clear race or ethnic background cannot be established, use Concept_Id 0. Accepted Race Concepts.	integer	Yes	No	Yes	CONCEPT	Race

ethnicity_concept_id	This field captures Ethnicity as defined by the Office of Management and Budget (OMB) of the US Government: it distinguishes only between “Hispanic” and “Not Hispanic”. Races and ethnic backgrounds are not stored here.	Only use this field if you have US-based data and a source of this information. Do not attempt to infer Ethnicity from the race or ethnic background of the Person. Accepted ethnicity concepts	integer	Yes	No	Yes	CONCEPT	Ethnicity
location_id	The location refers to the physical address of the person. This field should	Put the location_id from the LOCATION table here that represents the most granular location information for the person. For additional information on how to populate this field,	integer	No	No	Yes	LOCATION	

	capture the last known location of the person.	please refer to the THEMIS repository.					
provider_id	The Provider refers to the last known primary care provider (General Practitioner).	Put the provider_id from the PROVIDER table of the last known general practitioner of the person. If there are multiple providers, it is up to the ETL to decide which to put here.	integer	No	No	Yes	PROVIDER
care_site_id	The Care Site refers to where the Provider typically provides the primary care.		integer	No	No	Yes	CARE_SITE
person_source_value	Use this field to link back to persons in the source	Some use cases require the ability to link back to persons in the source data. This field allows for the storing of the person value as it appears in	varchar(50)	No	No	No	

	data. This is typically used for error checking of ETL logic.	the source. This field is not required but strongly recommended.					
gender_source_value	This field is used to store the biological sex of the person from the source data. It is not intended for use in standard analytics but for reference only.	Put the assigned sex at birth of the person as it appears in the source data.	varchar (50)	No	No	No	
gender_source_concept_id	Due to the small number of options, this tends to be zero.	If the source data codes asigned sex at birth in a non-standard vocabulary, store the concept_id here.	integer	No	No	Yes	CONCEPT

race_source_value	This field is used to store the race of the person from the source data. It is not intended for use in standard analytics but for reference only.	Put the race of the person as it appears in the source data.	varchar (50)	No	No	No	
race_source_concept_id	Due to the small number of options, this tends to be zero.	If the source data codes race in an OMOP supported vocabulary store the concept_id here.	integer	No	No	Yes	CONCEPT

ethnicity_source_value	This field is used to store the ethnicity of the person from the source data. It is not intended for use in standard analytics but for reference only.	If the person has an ethnicity other than the OMB standard of “Hispanic” or “Not Hispanic” store that value from the source data here.	varchar(50)	No	No	No	
ethnicity_source_concept_id	Due to the small number of options, this tends to be zero.	If the source data codes ethnicity in an OMOP supported vocabulary, store the concept_id here.	integer	No	No	Yes	CONCEPT

observation_period

Table Description

This table contains records which define spans of time during which two conditions are expected to hold: (i) Clinical Events that happened to the Person are recorded in the Event tables, and (ii) absence of records indicate such Events did not occur during this span of time.

User Guide

For each Person, one or more OBSERVATION_PERIOD records may be present, but they will not overlap or be back to back to each other. Events may exist outside all of the time spans of the OBSERVATION_PERIOD records for a patient, however, absence of an Event outside these time spans cannot be construed as evidence of absence of an Event. Incidence or prevalence rates should only be calculated for the time of active OBSERVATION_PERIOD records. When constructing cohorts, outside Events can be used for inclusion criteria definition, but without any guarantee for the performance of these criteria. Also, OBSERVATION_PERIOD records can be as short as a single day, greatly disturbing the denominator of any rate calculation as part of cohort characterizations. To avoid that, apply minimal observation time as a requirement for any cohort definition.

ETL Conventions

Each Person needs to have at least one OBSERVATION_PERIOD record, which should represent time intervals with a high capture rate of Clinical Events. Some source data have very similar concepts, such as enrollment periods in insurance claims data. In other source data such as most EHR systems these time spans need to be inferred under a set of assumptions. It is the discretion of the ETL developer to define these assumptions. In many ETL solutions the start date of the first occurrence or the first high quality occurrence of a Clinical Event (Condition, Drug, Procedure, Device, Measurement, Visit) is defined as the start of the OBSERVATION_PERIOD record, and the end date of the last occurrence of last high quality occurrence of a Clinical Event, or the end of the database period becomes the end of the OBSERVATOIN_PERIOD for each Person. If a Person only has a single Clinical Event the OBSERVATION_PERIOD record can be as short as one day. Depending on these definitions it is possible that Clinical Events fall outside the time spans defined by OBSERVATION_PERIOD records. Family history or history of Clinical Events generally are not used to generate OBSERVATION_PERIOD records around the time they are referring to. Any two overlapping or adjacent OBSERVATION_PERIOD records have to be merged into one.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
observation_period_id	A Person can have multiple discrete Observation Periods which are identified	Assign a unique observation_period_id to each discrete Observation Period for a Person.	integer	Yes	Yes	No		

	by the Observation_Period_Id.						
person_id	The Person ID of the PERSON record for which the Observation Period is recorded.		integer	Yes	No	Yes	PERSON
observation_period_start_date	Use this date to determine the start date of the Observation Period.	It is often the case that the idea of Observation Periods does not exist in source data. In those cases, the observation_period_start_date can be inferred as the earliest Event date available for the Person. In insurance claim data, the Observation Period can be considered as the time period the Person is enrolled with a payer. If a Person switches plans but stays with the same payer, and therefore capturing of data continues, that change would be captured in	date	Yes	No	No	

		PAYER_PLAN_PERIOD.						
observation_period_end_date	Use this date to determine the end date of the period for which we can assume that all events for a Person are recorded.	It is often the case that the idea of Observation Periods does not exist in source data. In those cases, the observation_period_end_date can be inferred as the last Event date available for the Person. In insurance claim data, the Observation Period can be considered as the time period the Person is enrolled with a payer.	date	Yes	No	No		
period_type_concept_id	This field can be used to determine the provenance of the Observation Period as in whether the period was determined from an insurance enrollment file, EHR healthcare encounters, or other sources.	Choose the observation_period_type_concept_id that best represents how the period was determined. Accepted Concepts. A more detailed explanation of each Type Concept can be found on the vocabulary wiki.	integer	Yes	No	Yes	CONCEPT	Type Concept

visit_occurrence

Table Description

This table contains Events where Persons engage with the healthcare system for a duration of time. They are often also called “Encounters”. Visits are defined by a configuration of circumstances under which they occur, such as (i) whether the patient comes to a healthcare institution, the other way around, or the interaction is remote, (ii) whether and what kind of trained medical staff is delivering the service during the Visit, and (iii) whether the Visit is transient or for a longer period involving a stay in bed.

User Guide

The configuration defining the Visit are described by Concepts in the Visit Domain, which form a hierarchical structure, but rolling up to generally familiar Visits adopted in most healthcare systems worldwide:

- Inpatient Visit: Person visiting hospital, at a Care Site, in bed, for duration of more than one day, with physicians and other Providers permanently available to deliver service around the clock
- Emergency Room Visit: Person visiting dedicated healthcare institution for treating emergencies, at a Care Site, within one day, with physicians and Providers permanently available to deliver service around the clock
- Emergency Room and Inpatient Visit: Person visiting ER followed by a subsequent Inpatient Visit, where Emergency department is part of hospital, and transition from the ER to other hospital departments is undefined
- Non-hospital institution Visit: Person visiting dedicated institution for reasons of poor health, at a Care Site, long-term or permanently, with no physician but possibly other Providers permanently available to deliver service around the clock
- Outpatient Visit: Person visiting dedicated ambulatory healthcare institution, at a Care Site, within one day, without bed, with physicians or medical Providers delivering service during Visit
- Home Visit: Provider visiting Person, without a Care Site, within one day, delivering service
- Telehealth Visit: Patient engages with Provider through communication media
- Pharmacy Visit: Person visiting pharmacy for dispensing of Drug, at a Care Site, within one day
- Laboratory Visit: Patient visiting dedicated institution, at a Care Site, within one day, for the purpose of a Measurement.

- Ambulance Visit: Person using transportation service for the purpose of initiating one of the other Visits, without a Care Site, within one day, potentially with Providers accompanying the Visit and delivering service
- Case Management Visit: Person interacting with healthcare system, without a Care Site, within a day, with no Providers involved, for administrative purposes

The Visit duration, or ‘length of stay’, is defined as VISIT_END_DATE - VISIT_START_DATE. For all Visits this is <1 day, except Inpatient Visits and Non-hospital institution Visits. The CDM also contains the VISIT_DETAIL table where additional information about the Visit is stored, for example, transfers between units during an inpatient Visit.

ETL Conventions

Visits can be derived easily if the source data contain coding systems for Place of Service or Procedures, like CPT codes for well visits. In those cases, the codes can be looked up and mapped to a Standard Visit Concept. Otherwise, Visit Concepts have to be identified in the ETL process. This table will contain concepts in the Visit domain. These concepts are arranged in a hierarchical structure to facilitate cohort definitions by rolling up to generally familiar Visits adopted in most healthcare systems worldwide. Visits can be adjacent to each other, i.e. the end date of one can be identical with the start date of the other. As a consequence, more than one-day Visits or their descendants can be recorded for the same day. Multi-day visits must not overlap, i.e. share days other than start and end days. It is often the case that some logic should be written for how to define visits and how to assign Visit_Concept_Id. For example, in US claims outpatient visits that appear to occur within the time period of an inpatient visit can be rolled into one with the same Visit_Occurrence_Id. In EHR data inpatient visits that are within one day of each other may be strung together to create one visit. It will all depend on the source data and how encounter records should be translated to visit occurrences. Providers can be associated with a Visit through the PROVIDER_ID field, or indirectly through PROCEDURE_OCCURRENCE records linked both to the VISIT and PROVIDER tables.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain

visit_occurrence_id	Use this to identify unique interactions between a person and the health care system. This identifier links across the other CDM event tables to associate events with a visit.	This should be populated by creating a unique identifier for each unique interaction between a person and the healthcare system where the person receives a medical good or service over a span of time.	integer	Yes	Yes	No		
person_id			integer	Yes	No	Yes	PERSON	
visit_concept_id	This field contains a concept id representing the kind of visit, like inpatient or outpatient. All concepts in this field should be standard and belong to the Visit domain.	Populate this field based on the kind of visit that took place for the person. For example this could be “Inpatient Visit”, “Outpatient Visit”, “Ambulatory Visit”, etc. This table will contain standard concepts in the Visit domain. These concepts are arranged in a hierarchical structure to	integer	Yes	No	Yes	CONCEPT	Visit

		facilitate cohort definitions by rolling up to generally familiar Visits adopted in most healthcare systems worldwide. Accepted Concepts.				
visit_start_date	For inpatient visits, the start date is typically the admission date. For outpatient visits the start date and end date will be the same.	When populating VISIT_START_DATE, you should think about the patient experience to make decisions on how to define visits. In the case of an inpatient visit this should be the date the patient was admitted to the hospital or institution. In all other cases this should be the date of the patient-provider interaction. If this information is not available the record should be dropped.	date	Yes	No	No

visit_start_datetime		If no time is given for the start date of a visit, set it to midnight (00:00:0000).	datetime	No	No	No
visit_end_date	For inpatient visits the end date is typically the discharge date. If a Person is still an inpatient in the hospital at the time of the data extract and does not have a visit_end_date, then set the visit_end_date to the date of the data pull.	Visit end dates are mandatory. If end dates are not provided in the source there are three ways in which to derive them: - Outpatient Visit: visit_end_datetime = visit_start_datetime - Emergency Room Visit: visit_end_datetime = visit_start_datetime - Inpatient Visit: Usually there is information about discharge. If not, you should be able to derive the end date from the sudden decline of activity or from the absence of inpatient procedures/drugs. - Non-hospital institution Visits:	date	Yes	No	No

		Particularly for claims data, if end dates are not provided assume the visit is for the duration of month that it occurs. For Inpatient Visits ongoing at the date of ETL, put date of processing the data into visit_end_datetime and visit_type_concept_id with 32220 “Still patient” to identify the visit as incomplete. - All other Visits: visit_end_datetime = visit_start_datetime. If this is a one-day visit the end date should match the start date.				
visit_end_datetime	If a Person is still an inpatient in the hospital at the time of the data extract and does not have a visit_end_datetime, then set the	If no time is given for the end date of a visit, set it to midnight (00:00:0000).	datetime	No	No	No

	visit_end_datetime to the datetime of the data pull.							
visit_type_concept_id	Use this field to understand the provenance of the visit record, or where the record comes from.	Populate this field based on the provenance of the visit record, as in whether it came from an EHR record or billing claim. [Accepted Concepts](). A more detailed explanation of each Type Concept can be found on the [vocabulary wiki]().	Integer	Yes	No	Yes	CONCEPT	Type Concept
provider_id	There will only be one provider per visit record and the ETL document should clearly state how they were chosen (attending, admitting, etc.). If there are multiple providers associated with a visit in the	If there are multiple providers associated with a visit, you will need to choose which one to put here. The additional providers can be stored in the [VISIT_DETAIL]() table.	integer	No	No	Yes	PROVIDER	

	source, this can be reflected in the event tables (CONDITION_OCCURRENCE, PROCEDURE_OCCURRENCE, etc.) or in the VISIT_DETAIL table.						
care_site_id	This field provides information about the Care Site where the Visit took place.	There should only be one Care Site associated with a Visit.	integer	No	No	Yes	CARE_SITE
visit_source_value	This field houses the verbatim value from the source data representing the kind of visit that took place (inpatient, outpatient, emergency, etc.)	If there is information about the kind of visit in the source data that value should be stored here. If a visit is an amalgamation of visits from the source then use a hierarchy to choose the visit source value, such as IP -> ER-> OP. This should line up with the logic chosen to	varchar(50)	No	No	No	

		determine how visits are created.						
visit_source_concept_id		If the visit source value is coded in the source data using an OMOP supported vocabulary put the concept id representing the source value here.	integer	No	No	Yes	CONCEPT	
admitted_from_concept_id	Use this field to determine where the patient was admitted from. This concept is part of the visit domain and can indicate if a patient was admitted to the hospital from a long-term care facility, for example.	If available, map the admitted_from_source_value to a standard concept in the visit domain. Accepted Concepts. If a person was admitted from home or was self-referred, set this to 0.	integer	No	No	Yes	CONCEPT	Visit

admitted_from_source_value		This information may be called something different in the source data but the field is meant to contain a value indicating where a person was admitted from. Typically this applies only to visits that have a length of stay, like inpatient visits or long-term care visits.	varchar(50)	No	No	No		
discharged_to_concept_id	Use this field to determine where the patient was discharged to after a visit. This concept is part of the visit domain and can indicate if a patient was transferred to another hospital or sent to a long-term care facility, for example. It is assumed that a person is	If available, map the discharged_to_source_value to a standard concept in the visit domain. Accepted Concepts.	integer	No	No	Yes	CONCEPT	Visit

	discharged to home therefore there is not a standard concept id for “home”. Use concept id = 0 when a person is discharged to home.						
discharged_to_source_value		This information may be called something different in the source data but the field is meant to contain a value indicating where a person was discharged to after a visit, as in they went home or were moved to long-term care. Typically this applies only to visits that have a length of stay of a day or more.	varchar(50)	No	No	No	
preceding_visit_occurrence_id	Use this field to find the visit that occurred for the person prior to the given visit. There could be	This field can be used to link a visit immediately preceding the current visit. Note this is not	integer	No	No	Yes	VISIT_OCCURRENCE

	a few days or a few years in between.	symmetrical, and there is no such thing as a “following_visit_id”.

visit_detail

Table Description

The VISIT_DETAIL table is an optional table used to represents details of each record in the parent VISIT_OCCURRENCE table. A good example of this would be the movement between units in a hospital during an inpatient stay or claim lines associated with a one insurance claim. For every record in the 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 visit domain.

User Guide

The configuration defining the Visit Detail is described by Concepts in the Visit Domain, which form a hierarchical structure. The Visit Detail record will have an associated to the Visit Occurrence record in two ways:
1. The Visit Detail record will have the VISIT_OCCURRENCE_ID it is associated to 2. The VISIT_DETAIL_CONCEPT_ID will be a descendant of the VISIT_CONCEPT_ID for the Visit.

ETL Conventions

It is not mandatory that the VISIT_DETAIL table be filled in, but if you find that the logic to create VISIT_OCCURRENCE records includes the roll-up of multiple smaller records to create one picture of a Visit then it is a good idea to use VISIT_DETAIL. In EHR data, for example, a Person may be in the hospital but instead of one over-arching Visit their encounters are recorded as times they interacted with a health care provider. A Person in the hospital interacts with multiple providers multiple times a day so the encounters must be strung together using some heuristic (defined by the ETL) to identify the entire Visit. In this case the encounters would be considered Visit Details and the entire Visit would be the Visit Occurrence. In this example it is also possible to use the Vocabulary to distinguish Visit Details from a Visit Occurrence by setting the VISIT_CONCEPT_ID to 9201 and the

VISIT_DETAIL_CONCEPT_IDs either to 9201 or its children to indicate where the patient was in the hospital at the time of care.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
visit_detail_id	Use this to identify unique interactions between a person and the health care system. This identifier links across the other CDM event tables to associate events with a visit detail.	This should be populated by creating a unique identifier for each unique interaction between a person and the healthcare system where the person receives a medical good or service over a span of time.	integer	Yes	Yes	No		
person_id			integer	Yes	No	Yes	PERSON	
visit_detail_concept_id	This field contains a concept id representing the kind of visit detail, like inpatient or outpatient. All concepts in this field should be standard and belong to the Visit domain.	Populate this field based on the kind of visit that took place for the person. For example this could be “Inpatient Visit”, “Outpatient Visit”, “Ambulatory Visit”, etc. This table will contain standard concepts	integer	Yes	No	Yes	CONCEPT	Visit

		in the Visit domain. These concepts are arranged in a hierarchical structure to facilitate cohort definitions by rolling up to generally familiar Visits adopted in most healthcare systems worldwide. Accepted Concepts.				
visit_detail_start_date	This is the date of the start of the encounter. This may or may not be equal to the date of the Visit the Visit Detail is associated with.	When populating VISIT_DETAIL_START_DATE, you should think about the patient experience to make decisions on how to define visits. Most likely this should be the date of the patient-provider interaction.	date	Yes	No	No
visit_detail_start_datetime		If no time is given for the start date of a visit, set it to midnight (00:00:0000).	datetime	No	No	No

visit_detail_end_date	This the end date of the patient-provider interaction. If a Person is still an inpatient in the hospital at the time of the data extract and does not have a visit_end_date, then set the visit_end_date to the date of the data pull.	Visit Detail end dates are mandatory. If end dates are not provided in the source there are three ways in which to derive them: - Outpatient Visit Detail: visit_detail_end_datetime = visit_detail_start_datetime - Emergency Room Visit Detail: visit_detail_end_datetime = visit_detail_start_datetime - Inpatient Visit Detail: Usually there is information about discharge. If not, you should be able to derive the end date from the sudden decline of activity or from the absence of inpatient procedures/drugs. - Non-hospital institution Visit Details:	date	Yes	No	No

		Particularly for claims data, if end dates are not provided assume the visit is for the duration of month that it occurs. For Inpatient Visit Details ongoing at the date of ETL, put date of processing the data into visit_detai_end_datetime and visit_detail_type_concept_id with 32220 “Still patient” to identify the visit as incomplete. All other Visits Details: visit_detail_end_datetime = visit_detail_start_datetime.						
visit_detail_end_datetime	If a Person is still an inpatient in the hospital at the time of the data extract and does not have a visit_end_datetime, then set the visit_end_datetime	If no time is given for the end date of a visit, set it to midnight (00:00:0000).	datetime	No	No	No		

	to the datetime of the data pull.							
visit_detail_type_concept_id	Use this field to understand the provenance of the visit detail record, or where the record comes from.	Populate this field based on the provenance of the visit detail record, as in whether it came from an EHR record or billing claim. Accepted Concepts. A more detailed explanation of each Type Concept can be found on the vocabulary wiki.	integer	Yes	No	Yes	CONCEPT	Type Concept
provider_id	There will only be one provider per **visit** record and the ETL document should clearly state how they were chosen (attending, admitting, etc.). This is a typical reason for leveraging the VISIT_DETAIL table as even though each VISIT_DETAIL record can only	The additional providers associated to a Visit can be stored in this table where each VISIT_DETAIL record represents a different provider.	integer	No	No	Yes	PROVIDER	

	have one provider, there is no limit to the number of VISIT_DETAIL records that can be associated to a VISIT_OCCURRENCE record.						
care_site_id	This field provides information about the Care Site where the Visit Detail took place.	There should only be one Care Site associated with a Visit Detail.	integer	No	No	Yes	CARE_SITE
visit_detail_source_value	This field houses the verbatim value from the source data representing the kind of visit detail that took place (inpatient, outpatient, emergency, etc.)	If there is information about the kind of visit detail in the source data that value should be stored here. If a visit is an amalgamation of visits from the source then use a hierarchy to choose the VISIT_DETAIL_SOURCE_VALUE, such as IP -> ER-> OP. This should line up with the logic chosen to determine how visits are created.	varchar(50)	No	No	No	

visit_detail_source_concept_id		If the VISIT_DETAIL_SOURCE_VALUE is coded in the source data using an OMOP supported vocabulary put the concept id representing the source value here.	Integer	No	No	Yes	CONCEPT	
admitted_from_concept_id	Use this field to determine where the patient was admitted from. This concept is part of the visit domain and can indicate if a patient was admitted to the hospital from a long-term care facility, for example.	If available, map the admitted_from_source_value to a standard concept in the visit domain. Accepted Concepts. If a person was admitted from home or was self-referred, set this to 0.	Integer	No	No	Yes	CONCEPT	Visit
admitted_from_source_value		This information may be called something different in the source data but the field is meant to contain a value indicating where a person was admitted from. Typically this	varchar(50)	No	No	No		

	applies only to visits that have a length of stay, like inpatient visits or long-term care visits.				
discharged_to_source_value	This information may be called something different in the source data but the field is meant to contain a value indicating where a person was discharged to after a visit, as in they went home or were moved to long-term care. Typically this applies only to visits that have a length of stay of a day or more.	varchar(50)	No	No	No

discharged_to_concept_id	Use this field to determine where the patient was discharged to after a visit. This concept is part of the visit domain and can indicate if a patient was transferred to another hospital or sent to a long-term care facility, for example. It is assumed that a person is discharged to home therefore there is not a standard concept id for “home”. Use concept id = 0 when a person is discharged to home.	If available, map the DISCHARGE_TO_SOURCE_VALUE to a Standard Concept in the Visit domain. Accepted Concepts.	integer	No	No	Yes	CONCEPT	Visit
preceding_visit_detail_id	Use this field to find the visit detail that occurred for the person prior to the given visit detail record. There could be a few days or a few years in between.	The PRECEDING_VISIT_DETAIL_ID can be used to link a visit immediately preceding the current Visit Detail. Note this is not symmetrical, and there is no such thing as a	integer	No	No	Yes	VISIT_DETAIL	

		“following_visit_id” .					
parent_visit_detail_id	Use this field to find the visit detail that subsumes the given visit detail record. This is used in the case that a visit detail record needs to be nested beyond the VISIT_OCCURRENCE/VISIT_DETAIL relationship.	If there are multiple nested levels to how Visits are represented in the source, the VISIT_DETAIL_PARENT_ID can be used to record this relationship.	integer	No	No	Yes	VISIT_DETAIL
visit_occurrence_id	Use this field to link the VISIT_DETAIL record to its VISIT_OCCURRENCE.	Put the VISIT_OCCURRENCE_ID that subsumes the VISIT_DETAIL record here.	integer	Yes	No	Yes	VISIT_OCCURRENCE

condition_occurrence

Table Description

This table contains records of Events 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.

User Guide

Conditions are defined by Concepts from the Condition domain, which form a complex hierarchy. As a result, the same Person with the same disease may have multiple Condition records, which belong to the same hierarchical family. Most Condition records are mapped from diagnostic codes, but recorded signs, symptoms and summary descriptions also contribute to this table. Rule out diagnoses should not be recorded in this table, but in reality their negating nature is not always captured in the source data, and other precautions must be taken when when identifying Persons who should suffer from the recorded Condition. Record all conditions as they exist in the source data. Any decisions about diagnosis/phenotype definitions would be done through cohort specifications. These cohorts can be housed in the COHORT table. Conditions span a time interval from start to end, but are typically recorded as single snapshot records with no end date. The reason is twofold: (i) At the time of the recording the duration is not known and later not recorded, and (ii) the Persons typically cease interacting with the healthcare system when they feel better, which leads to incomplete capture of resolved Conditions. The CONDITION_ERA table addresses this issue. Family history and past diagnoses ('history of') are not recorded in this 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', should not represented here. Instead, they should be recorded in the OBSERVATION table, if they are used for analyses. However, this information is not always available.

ETL Conventions

Source codes and source text fields mapped to Standard Concepts of the Condition Domain have to be recorded here.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
condition_occurrence_id	The unique key given to a condition record for a person. Refer to the ETL for how duplicate conditions during the same visit were handled.	Each instance of a condition present in the source data should be assigned this unique key. In some cases, a person can have multiple records of the same condition within the same visit. It is valid to keep these duplicates and assign them	integer	Yes	Yes	No		

		individual, unique, CONDITION_OCCURRENCE_IDs, though it is up to the ETL how they should be handled.						
person_id	The PERSON_ID of the PERSON for whom the condition is recorded.		integer	Yes	No	Yes	PERSON	
condition_concept_id	The CONDITION_CONCEPT_ID field is recommended for primary use in analyses, and must be used for network studies. This is the standard concept mapped from the source value which represents a condition	The CONCEPT_ID that the CONDITION_SOURCE_VALUE maps to. Only records whose source values map to concepts with a domain of “Condition” should go in this table. Accepted Concepts.	integer	Yes	No	Yes	CONCEPT	Condition

condition_start_date	Use this date to determine the start date of the condition	Most often data sources do not have the idea of a start date for a condition. Rather, if a source only has one date associated with a condition record it is acceptable to use that date for both the CONDITION_START_DATE and the CONDITION_END_DATE.	date	Yes	No	No
condition_start_datetime		If a source does not specify datetime the convention is to set the time to midnight (00:00:0000)	datetime	No	No	No
condition_end_date	Use this date to determine the end date of the condition	Most often data sources do not have the idea of a start date for a condition. Rather, if a source only has one date associated with a condition record it is acceptable to use that date for both the CONDITION_START_DATE and the	date	No	No	No

		CONDITION_END_DATE.						
condition_end_datetime		If a source does not specify datetime the convention is to set the time to midnight (00:00:0000)	datetime	No	No	No		
condition_type_concept_id	This field can be used to determine the provenance of the Condition record, as in whether the condition was from an EHR system, insurance claim, registry, or other sources.	Choose the CONDITION_TYPE_CONCEPT_ID that best represents the provenance of the record. Accepted Concepts. A more detailed explanation of each Type Concept can be found on the vocabulary wiki.	integer	Yes	No	Yes	CONCEPT	Type Concept
condition_status_concept_id	This concept represents the point during the visit the diagnosis was given (admitting diagnosis, final diagnosis), whether the diagnosis was determined due to	Choose the Concept in the Condition Status domain that best represents the point during the visit when the diagnosis was given. These can include admitting diagnosis, principal diagnosis, and secondary diagnosis.	integer	No	No	Yes	CONCEPT	Condition Status

	laboratory findings, if the diagnosis was exclusionary, or if it was a preliminary diagnosis, among others.	Accepted Concepts.					
stop_reason	The Stop Reason indicates why a Condition is no longer valid with respect to the purpose within the source data. Note that a Stop Reason does not necessarily imply that the condition is no longer occurring.	This information is often not populated in source data and it is a valid etl choice to leave it blank if the information does not exist.	varchar(20)	No	No	No	
provider_id	The provider associated with condition record, e.g. the provider who made the diagnosis or the provider	The ETL may need to make a choice as to which PROVIDER_ID to put here. Based on what is available this may or may not be different than the provider associated with the overall	integer	No	No	Yes	PROVIDER

	who recorded the symptom.	VISIT_OCCURRENCE record, for example the admitting vs attending physician on an EHR record.					
visit_occurrence_id	The visit during which the condition occurred.	Depending on the structure of the source data, this may have to be determined based on dates. If a CONDITION_START_DATE occurs within the start and end date of a Visit it is a valid ETL choice to choose the VISIT_OCCURRENCE_ID from the Visit that subsumes it, even if not explicitly stated in the data. While not required, an attempt should be made to locate the VISIT_OCCURRENCE_ID of the CONDITION_OCCURRENCE record.	integer	No	No	Yes	VISIT_OCCURRENCE

visit_detail_id	The VISIT_DETAIL record during which the condition occurred. For example, if the person was in the ICU at the time of the diagnosis the VISIT_OCCURRENCE record would reflect the overall hospital stay and the VISIT_DETAIL record would reflect the ICU stay during the hospital visit.	Same rules apply as for the VISIT_OCCURRENCE_ID.	integer	No	No	Yes	VISIT_DETAIL
condition_source_value	This field houses the verbatim value from the source data representing the condition that occurred. For example, this could be an ICD10 or Read code.	This code is mapped to a Standard Condition Concept in the Standardized Vocabularies and the original code is stored here for reference.	varchar(50)	No	No	No	

condition_source_concept_id	This is the concept representing the condition source value and may not necessarily be standard. This field is discouraged from use in analysis because it is not required to contain Standard Concepts that are used across the OHDSI community, and should only be used when Standard Concepts do not adequately represent the source detail for the Condition necessary for a given analytic use case. Consider using CONDITION_CONCEPT_ID	If the CONDITION_SOURCE_VALUE is coded in the source data using an OMOP supported vocabulary put the concept id representing the source value here.	integer	No	No	Yes	CONCEPT

	instead to enable standardized analytics that can be consistent across the network.					
condition_status_source_value	This field houses the verbatim value from the source data representing the condition status.	This information may be called something different in the source data but the field is meant to contain a value indicating when and how a diagnosis was given to a patient. This source value is mapped to a standard concept which is stored in the CONDITION_STATUS_CONCEPT_ID field.	varchar(50)	No	No	No

drug_exposure

Table Description

This table captures records about the exposure to a Drug 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 biochemical effect on the metabolism. 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.

User Guide

The purpose of records in this table is to indicate an exposure to a certain drug as best as possible. In this context a drug is defined as an active ingredient. Drug Exposures are defined by Concepts from the Drug domain, which form a complex hierarchy. As a result, one DRUG_SOURCE_CONCEPT_ID may map to multiple standard concept ids if it is a combination product. Records in this table represent prescriptions written, prescriptions dispensed, and drugs administered by a provider to name a few. The DRUG_TYPE_CONCEPT_ID can be used to find and filter on these types. This table includes additional information about the drug products, the quantity given, and route of administration.

ETL Conventions

Information about quantity and dose is provided in a variety of different ways and it is important for the ETL to provide as much information as possible from the data. Depending on the provenance of the data fields may be captured differently i.e. quantity for drugs administered may have a separate meaning from quantity for prescriptions dispensed. If a patient has multiple records on the same day for the same drug or procedures the ETL should not de-dupe them unless there is probable reason to believe the item is a true data duplicate. Take note on how to handle refills for prescriptions written.

For detailed conventions on how to populate this table, please refer to the THEMIS repository.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain

drug_exposure_id	The unique key given to records of drug dispensings or administrations for a person. Refer to the ETL for how duplicate drugs during the same visit were handled.	Each instance of a drug dispensing or administration present in the source data should be assigned this unique key. In some cases, a person can have multiple records of the same drug within the same visit. It is valid to keep these duplicates and assign them individual, unique, DRUG_EXPOSURE_IDs, though it is up to the ETL how they should be handled.	integer	Yes	Yes	No	
person_id	The PERSON_ID of the PERSON for whom the drug dispensing or administration is recorded. This may be a system generated code.		integer	Yes	No	Yes	PERSON

drug_concept_id	The DRUG_CONCEPT_ID field is recommended for primary use in analyses, and must be used for network studies. This is the standard concept mapped from the source concept id which represents a drug product or molecule otherwise introduced to the body. The drug concepts can have a varying degree of information about drug strength and dose. This information is relevant in the context of quantity and administration information in the subsequent fields plus strength information from the DRUG_STREN	The CONCEPT_ID that the DRUG_SOURCE_VALUE maps to. The concept id should be derived either from mapping from the source concept id or by picking the drug concept representing the most amount of detail you have. Records whose source values map to standard concepts with a domain of Drug should go in this table. 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	integer	Yes	No	Yes	CONCEPT	Drug

	GTH table, provided as part of the standard vocabulary download.	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: Marketed Product, Branded Pack, Clinical Pack, Branded Drug, Clinical Drug, Branded Drug Component, Clinical Drug Component, Branded Drug Form, Clinical Drug Form, and only if no other information is available Ingredient. Note: If only the drug class is known, the DRUG_CONCEPT_ID field should contain 0.

		Accepted Concepts.				
drug_exposure_start_date	Use this date to determine the start date of the drug record.	Valid entries include a start date of a prescription, the date a prescription was filled, or the date on which a Drug administration was recorded. It is a valid ETL choice to use the date the drug was ordered as the DRUG_EXPOSURE_START_DATE.	date	Yes	No	No
drug_exposure_start_datetime		This is not required, though it is in v6. If a source does not specify datetime the convention is to set the time to midnight (00:00:0000)	datetime	No	No	No

drug_exposure_end_date	The DRUG_EXPOSURE_END_DATE denotes the day the drug exposure ended for the patient.	If this information is not explicitly available in the data, infer the end date using the following methods: 1. Start first with duration or days supply using the calculation drug start date + days supply -1 day. 2. Use quantity divided by daily dose that you may obtain from the sig or a source field (or assumed daily dose of 1) for solid, indivisibile, drug products. If quantity represents ingredient amount, quantity divided by daily dose * concentration (from drug_strength) drug concept id tells you the dose form. 3. If it is an	date	Yes	No	No

administration record, set drug end date equal to drug start date. If the record is a written prescription then set end date to start date + 29. If the record is a mail-order prescription set end date to start date + 89. The end date must be equal to or greater than the start date. Ibuprofen 20mg/mL oral solution concept tells us this is oral solution. Calculate duration as quantity (200 example) * daily dose (5mL) /concentration (20mg/mL) 200*5/20 = 50 days. Examples by dose form

For detailed conventions for how to populate this field, please

		see the THEMIS repository.						
drug_exposure_end_datetime		This is not required, though it is in v6. If a source does not specify datetime the convention is to set the time to midnight (00:00:0000)	datetime	No	No	No		
verbatim_end_date	This is the end date of the drug exposure as it appears in the source data, if it is given	Put the end date or discontinuation date as it appears from the source data or leave blank if unavailable.	date	No	No	No		
drug_type_concept_id	You can use the TYPE_CONCEPT_ID to delineate between prescriptions written vs. prescriptions dispensed vs. medication history vs. patient-reported exposure, etc.	Choose the drug_type_concept_id that best represents the provenance of the record, for example whether it came from a record of a prescription written or physician administered drug. Accepted Concepts. A more detailed	integer	Yes	No	Yes	CONCEPT	Type Concept

		explanation of each Type Concept can be found on the vocabulary wiki.				
stop_reason	The reason a person stopped a medication as it is represented in the source. Reasons include regimen completed, changed, removed, etc. This field will be retired in v6.0.	This information is often not populated in source data and it is a valid etl choice to leave it blank if the information does not exist.	varchar (20)	No	No	No
refills	This is only filled in when the record is coming from a prescription written this field is meant to represent intended refills at time of the prescription.		integer	No	No	No

quantity		To find the dose form of a drug the RELATIONSHIP table can be used where the relationship_id is ‘Has dose form’. If liquid, quantity stands for the total amount dispensed or ordered of ingredient in the units given by the drug_strength table. If the unit from the source data does not align with the unit in the DRUG_STRENGTH table the quantity should be converted to the correct unit given in DRUG_STRENGTH. For clinical drugs with fixed dose forms (tablets etc.) the quantity is the number of units/tablets/capsules prescribed or dispensed (can be partial, but then only 1/2 or 1/3,	float	No	No	No

not 0.01). Clinical drugs with divisible dose forms (injections) the quantity is the amount of ingredient the patient got. For example, if the injection is 2mg/mL but the patient got 80mL then quantity is reported as 160. Quantified clinical drugs with divisible dose forms (prefilled syringes), the quantity is the amount of ingredient similar to clinical drugs. Please see how to calculate drug dose for more information.

days_supply	The number of days of supply of the medication as recorded in the original prescription or dispensing record. Days supply can differ from actual drug duration (i.e. prescribed days supply vs actual exposure).”,“The field should be left empty if the source data does not contain a verbatim days_supply, and should not be calculated from other fields. Negative values are not allowed. If the source has negative days supply the record should be dropped as it is unknown if the patient actually took the drug. Several actions are possible: 1) record is not	integer	No	No	No

		trustworthy and we remove the record entirely. 2) we trust the record and leave days_supply empty or 3) record needs to be combined with other record (e.g. reversal of prescription). High values (>365 days) should be investigated. If considered an error in the source data (e.g. typo), the value needs to be excluded to prevent creation of unrealistic long eras.						
sig	This is the verbatim instruction for the drug as written by the provider.	Put the written out instructions for the drug as it is verbatim in the source, if available.	varchar (MAX)	No	No	No		
route_concept_id		The standard CONCEPT_ID that the ROUTE_SOURCE_VALUE maps	integer	No	No	Yes	CONCEPT	Route

		to in the route domain.					
lot_number			varchar (50)	No	No	No	
provider_id	The Provider associated with drug record, e.g. the provider who wrote the prescription or the provider who administered the drug.	The ETL may need to make a choice as to which PROVIDER_ID to put here. Based on what is available this may or may not be different than the provider associated with the overall VISIT_OCCURRENCE record, for example the ordering vs administering physician on an EHR record.	integer	No	No	Yes	PROVIDER
visit_occurrence_id	The Visit during which the drug was prescribed, administered or dispensed.	To populate this field drug exposures must be explicitly initiated in the visit.	integer	No	No	Yes	VISIT_OCCURRENCE

visit_detail_id	The VISIT_DETAIL record during which the drug exposure occurred. For example, if the person was in the ICU at the time of the drug administration the VISIT_OCCURRENCE record would reflect the overall hospital stay and the VISIT_DETAIL record would reflect the ICU stay during the hospital visit.	Same rules apply as for the VISIT_OCCURRENCE_ID.	integer	No	No	Yes	VISIT_DETAIL
drug_source_value	This field houses the verbatim value from the source data representing the drug exposure that occurred. For example, this could be an NDC or Gemscript code.	This code is mapped to a Standard Drug Concept in the Standardized Vocabularies and the original code is stored here for reference.	varchar(50)	No	No	No	

drug_source_concept_id	This is the concept representing the drug source value and may not necessarily be standard. This field is discouraged from use in analysis because it is not required to contain Standard Concepts that are used across the OHDSI community, and should only be used when Standard Concepts do not adequately represent the source detail for the Drug necessary for a given analytic use case. Consider using DRUG_CONCEPT_ID instead to enable standardized analytics that can be consistent	If the DRUG_SOURCE_VALUE is coded in the source data using an OMOP supported vocabulary put the concept id representing the source value here.	integer	No	No	Yes	CONCEPT

	across the network.					
route_source_value	This field houses the verbatim value from the source data representing the drug route.	This information may be called something different in the source data but the field is meant to contain a value indicating when and how a drug was given to a patient. This source value is mapped to a standard concept which is stored in the ROUTE_CONCEPT_ID field.	varchar (50)	No	No	No
dose_unit_source_value	This field houses the verbatim value from the source data representing the dose unit of the drug given.	This information may be called something different in the source data but the field is meant to contain a value indicating the unit of dosage of drug given to the patient. **This is an older column and will be deprecated in an**	varchar (50)	No	No	No

upcoming version.

procedure_occurrence

Table Description

This table contains records of activities or processes ordered by, or carried out by, a healthcare provider on the patient with a diagnostic or therapeutic purpose.

User Guide

Lab tests are not a procedure, if something is observed with an expected resulting amount and unit then it should be a measurement. Phlebotomy is a procedure but so trivial that it tends to be rarely captured. It can be assumed that there is a phlebotomy procedure associated with many lab tests, therefore it is unnecessary to add them as separate procedures. If the user finds the same procedure over concurrent days, it is assumed those records are part of a procedure lasting more than a day. This logic is in lieu of the procedure_end_date, which will be added in a future version of the CDM.

ETL Conventions

When dealing with duplicate records, the ETL must determine whether to sum them up into one record or keep them separate. Things to consider are: - Same Procedure - Same PROCEDURE_DATETIME - Same Visit Occurrence or Visit Detail - Same Provider - Same Modifier for Procedures. Source codes and source text fields mapped to Standard Concepts of the Procedure Domain have to be recorded here.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain

procedure_occurrence_id	The unique key given to a procedure record for a person. Refer to the ETL for how duplicate procedures during the same visit were handled.	Each instance of a procedure occurrence in the source data should be assigned this unique key. In some cases, a person can have multiple records of the same procedure within the same visit. It is valid to keep these duplicates and assign them individual, unique, PROCEDURE_OCCURRENCE_IDs, though it is up to the ETL how they should be handled.	integer	Yes	Yes	No	
person_id	The PERSON_ID of the PERSON for whom the procedure is recorded. This may be a system generated code.		integer	Yes	No	Yes	PERSON

procedure_concept_id	The PROCEDURE_CONCEPT_ID field is recommended for primary use in analyses, and must be used for network studies. This is the standard concept mapped from the source value which represents a procedure	The CONCEPT_ID that the PROCEDURE_SOURCE_VALUE maps to. Only records whose source values map to standard concepts with a domain of “Procedure” should go in this table. Accepted Concepts.	integer	Yes	No	Yes	CONCEPT	Procedure
procedure_date	Use this date to determine the date the procedure started.	This is meant to be the **start date** of the procedure. It will be renamed in a future version to **PROCEDURE_START_DATE**.	date	Yes	No	No		
procedure_datetime		If the procedure has a start time in the native date, use this field to house that information. This will be renamed in a future version to **PROCEDURE_START_DATETIME**.	datetime	No	No	No		

procedure_end_date	Use this field to house the date that the procedure ended.	This is meant to be the end date of the procedure. It is not required and for most cases will be the same as the PROCEDURE_START_DATE.	date	No	No	No		
procedure_end_datetime	Use this field to house the datetime that the procedure ended.	This is meant to house the end datetime of the procedure and will most often be used in conjunction with the procedure_start_datetime to determine the length of the procedure.	datetime	No	No	No		
procedure_type_concept_id	This field can be used to determine the provenance of the Procedure record, as in whether the procedure was from an EHR system, insurance claim, registry, or other sources.	Choose the PROCEDURE_TYPE_CONCEPT_ID that best represents the provenance of the record, for example whether it came from an EHR record or billing claim. If a procedure is recorded as an EHR encounter, the PROCEDURE_TYPE_CONCEPT would be 'EHR encounter record'. Accepted	integer	Yes	No	Yes	CONCEPT	Type Concept

		Concepts. A more detailed explanation of each Type Concept can be found on the vocabulary wiki.					
modifier_concept_id	The modifiers are intended to give additional information about the procedure but as of now the vocabulary is under review.	It is up to the ETL to choose how to map modifiers if they exist in source data. These concepts are typically distinguished by 'Modifier' concept classes (e.g., 'CPT4 Modifier' as part of the 'CPT4' vocabulary). If there is more than one modifier on a record, one should be chosen that pertains to the procedure rather than provider. Accepted Concepts.	integer	No	No	Yes	CONCEPT
quantity	If the quantity value is omitted, a single procedure is assumed.	If a Procedure has a quantity of '0' in the source, this should default to '1' in the ETL. If there is a record in the source it can be assumed the exposure	integer	No	No	No	

		occurred at least once					
provider_id	The provider associated with the procedure record, e.g. the provider who performed the Procedure.	The ETL may need to make a choice as to which PROVIDER_ID to put here. Based on what is available this may or may not be different than the provider associated with the overall VISIT_OCCURRENCE record, for example the admitting vs attending physician on an EHR record.	integer	No	No	Yes	PROVIDER
visit_occurrence_id	The visit during which the procedure occurred.	Depending on the structure of the source data, this may have to be determined based on dates. If a PROCEDURE_DATE occurs within the start and end date of a Visit it is a valid ETL choice to choose the VISIT_OCCURRENCE_ID from the Visit that subsumes it, even if not explicitly stated in the data.	integer	No	No	Yes	VISIT_OCCURRENCE

		While not required, an attempt should be made to locate the VISIT_OCCURRENCE_ID of the PROCEDURE_OCCURRENCE record.					
visit_detail_id	The VISIT_DETAIL record during which the Procedure occurred. For example, if the Person was in the ICU at the time of the Procedure the VISIT_OCCURRENCE record would reflect the overall hospital stay and the VISIT_DETAIL record would reflect the ICU stay during the hospital visit.	Same rules apply as for the VISIT_OCCURRENCE_ID.	integer	No	No	Yes	VISIT_DETAIL
procedure_source_value	This field houses the verbatim value from the source data	Use this value to look up the source concept id and then map the source	varchar(50)	No	No	No	

	representing the procedure that occurred. For example, this could be an CPT4 or OPCS4 code.	concept id to a standard concept id.						
procedure_source_concept_id	This is the concept representing the procedure source value and may not necessarily be standard. This field is discouraged from use in analysis because it is not required to contain Standard Concepts that are used across the OHDSI community, and should only be used when Standard Concepts do not adequately represent the source detail for the Procedure	If the PROCEDURE_SOURCE_VALUE is coded in the source data using an OMOP supported vocabulary put the concept id representing the source value here.	integer	No	No	Yes	CONCEPT	

	necessary for a given analytic use case. Consider using PROCEDURE_CONCEPT_ID instead to enable standardized analytics that can be consistent across the network.					
modifier_source_value	This field houses the verbatim value from the source data representing the modifier code for the procedure that occurred.	The value stored here is mapped to a MODIFIER_CONCEPT_ID.	varchar(50)	No	No	No

device_exposure

Table Description

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

User Guide

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.

ETL Conventions

Source codes and source text fields mapped to Standard Concepts of the Device Domain have to be recorded here.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
device_exposure_id	The unique key given to records a person's exposure to a foreign physical object or instrument.	Each instance of an exposure to a foreign object or device present in the source data should be assigned this unique key.	integer	Yes	Yes	No		
person_id			integer	Yes	No	Yes	PERSON	
device_concept_id	The DEVICE_CONCEPT_ID field is recommended for primary use in analyses, and must be used for network studies.	The CONCEPT_ID that the DEVICE_SOURCE_VALUE maps to.	integer	Yes	No	Yes	CONCEPT	Device

	This is the standard concept mapped from the source concept id which represents a foreign object or instrument the person was exposed to.					
device_exposure_start_date	Use this date to determine the start date of the device record.	Valid entries include a start date of a procedure to implant a device, the date of a prescription for a device, or the date of device administration.	date	Yes	No	No
device_exposure_start_datetime		This is not required, though it is in v6. If a source does not specify datetime the convention is to set the time to midnight (00:00:0000)	datetime	No	No	No

device_exposure_end_date	The DEVICE_EXPOSURE_END_DATE denotes the day the device exposure ended for the patient, if given.	Put the end date or discontinuation date as it appears from the source data or leave blank if unavailable.	date	No	No	No		
device_exposure_end_datetime		If a source does not specify datetime the convention is to set the time to midnight (00:00:0000)	datetime	No	No	No		
device_type_concept_id	You can use the TYPE_CONCEPT_ID to denote the provenance of the record, as in whether the record is from administrative claims or EHR.	Choose the drug_type_concept_id that best represents the provenance of the record, for example whether it came from a record of a prescription written or physician administered drug. Accepted	integer	Yes	No	Yes	CONCEPT	Type Concept

		Concepts. A more detailed explanation of each Type Concept can be found on the vocabulary wiki.				
unique_device_id	This is the Unique Device Identification (UDI-DI) number for devices regulated by the FDA, if given.	For medical devices that are regulated by the FDA, a Unique Device Identification (UDI) is provided if available in the data source and is recorded in the UNIQUE_DEVICE_ID field.	varchar(255)	No	No	No
production_id	This is the Production Identifier (UDI-PI) portion of the Unique Device Identification.		varchar(255)	No	No	No
quantity		If there is a record of device	integer	No	No	No

		exposure in the source but no quantity value, then set to 1.					
provider_id	The Provider associated with device record, e.g. the provider who wrote the prescription or the provider who implanted the device.	The ETL may need to make a choice as to which PROVIDER_ID to put here. Based on what is available this may or may not be different than the provider associated with the overall VISIT_OCCURRENCE record.	integer	No	No	Yes	PROVIDER
visit_occurrence_id	The Visit during which the device was prescribed or given.	To populate this field device exposures must be explicitly initiated in the visit.	integer	No	No	Yes	VISIT_OCCURRENCE

visit_detail_id	The Visit Detail during which the device was prescribed or given.	To populate this field device exposures must be explicitly initiated in the visit detail record.	integer	No	No	Yes	VISIT_DETAIL
device_source_value	This field houses the verbatim value from the source data representing the device exposure that occurred. For example, this could be an NDC or Gemscript code.	This code is mapped to a Standard Device Concept in the Standardized Vocabularies and the original code is stored here for reference.	varchar(50)	No	No	No	

device_source_concept_id	This is the concept representing the device source value and may not necessarily be standard. This field is discouraged from use in analysis because it is not required to contain Standard Concepts that are used across the OHDSI community, and should only be used when Standard Concepts do not adequately represent the source detail for the Device necessary for a given analytic use case. Consider using DEVICE_CONCEPT_ID instead to enable standardized analytics that can be consistent across the network.	If the DEVICE_SOURCE_VALUE is coded in the source data using an OMOP supported vocabulary put the concept id representing the source value here.	integer	No	No	Yes	CONCEPT

unit_concept_id	UNIT_SOURCE_VALUES should be mapped to a Standard Concept in the Unit domain that best represents the unit as given in the source data.	There is no standardization requirement for units associated with DEVICE_CONCEPT_IDs, however, it is the responsibility of the ETL to choose the most plausible unit. If the source unit is NULL (applicable to cases when there’s no numerical value or when it doesn’t require a unit), keep unit_concept_id NULL as well. If there’s no mapping of a source unit, populate unit_concept_id with 0.	integer	No	No	Yes	CONCEPT	Unit

unit_source_value	This field houses the verbatim value from the source data representing the unit of the Device. For example, blood transfusions are considered devices and can be given in mL quantities.	This code is mapped to a Standard Condition Concept in the Standardized Vocabularies and the original code is stored here for reference. Using the blood transfusion example, blood transfusion is represented by the DEVICE_CONCEPT_ID and the unit (mL) would be housed in the UNIT_SOURCE_VALUE and mapped to a standard concept in the unit domain.	varchar(50)	No	No	No

unit_source_concept_id	This is the concept representing the UNIT_SOURCE_VALUE and may not necessarily be standard. This field is discouraged from use in analysis because it is not required to contain Standard Concepts that are used across the OHDSI community, and should only be used when Standard Concepts do not adequately represent the source detail for the Unit necessary for a given analytic use case. Consider using UNIT_CONCEPT_ID instead to enable standardized analytics that can be consistent across the network.	If the UNIT_SOURCE_VALUE is coded in the source data using an OMOP supported vocabulary put the concept id representing the source value here.	integer	No	No	Yes	CONCEPT

measurement

Table Description

The MEASUREMENT table contains records of Measurements, 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. 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). The Procedure for obtaining the sample is housed in the PROCEDURE_OCCURRENCE table, though it is unnecessary to create a PROCEDURE_OCCURRENCE record for each measurement if one does not exist in the source data. Measurements differ from Observations in that they require a standardized test or some other activity to generate a quantitative or qualitative result. If there is no result, it is assumed that the lab test was conducted but the result was not captured.

User Guide

Measurements are predominately lab tests with a few exceptions, like blood pressure or function tests. Results are given in the form of a value and unit combination. When investigating measurements, look for operator_concept_ids (<, >, etc.).

ETL Conventions

Only records where the source value maps to a Concept in the measurement domain should be included in this table. Even though each Measurement always has a result, the fields VALUE_AS_NUMBER and VALUE_AS_CONCEPT_ID are not mandatory as often the result is not given in the source data. 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. For some Measurement Concepts, the result is included in the test. For example, ICD10 CONCEPT_ID 45548980 ‘Abnormal level of unspecified serum enzyme’ indicates a Measurement and the result (abnormal). 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 4046263 ‘Enzyme measurement’ as well as a ‘Maps to value’ record to 4135493 ‘Abnormal’.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
measurement_id	The unique key given to a Measurement record for a Person. Refer to the ETL for how duplicate Measurements during the same Visit were handled.	Each instance of a measurement present in the source data should be assigned this unique key. In some cases, a person can have multiple records of the same measurement within the same visit. It is valid to keep these duplicates and assign them individual, unique, MEASUREMENT_IDs, though it is up to the ETL how they should be handled.	integer	Yes	Yes	No		
person_id	The PERSON_ID of the Person for whom the Measurement is recorded. This may be a system generated code.		integer	Yes	No	Yes	PERSON	

measurement_concept_id	The MEASUREMENT_CONCEPT_ID field is recommended for primary use in analyses, and must be used for network studies. This is the standard concept mapped from the source value which represents a measurement.	The CONCEPT_ID that the MEASUREMENT_SOURCE_VALUE maps to. Only records whose source values map to concepts with a domain of Measurement should go in this table.	integer	Yes	No	Yes	CONCEPT	Measurement
measurement_date	Use this date to determine the date of the measurement.	If there are multiple dates in the source data associated with a record such as order_date, draw_date, and result_date, choose the one that is closest to the date the sample was drawn from the patient.	date	Yes	No	No		
measurement_datetime		This is not required, though it is in v6. If a source does not specify datetime the convention is to set the time to	datetime	No	No	No		

		midnight (00:00:0000)						
measurement_time		This is present for backwards compatibility and will be deprecated in an upcoming version.	varchar(10)	No	No	No		
measurement_type_concept_id	This field can be used to determine the provenance of the Measurement record, as in whether the measurement was from an EHR system, insurance claim, registry, or other sources.	Choose the MEASUREMENT_TYPE_CONCEPT_ID that best represents the provenance of the record, for example whether it came from an EHR record or billing claim. Accepted Concepts. A more detailed explanation of each Type Concept can be found on the vocabulary wiki.	integer	Yes	No	Yes	CONCEPT	Type Concept
operator_concept_id	The meaning of Concept 4172703 for '=' is identical to omission of a OPERATOR_CONCEPT_ID value. Since the	Operators are =, > and these concepts belong to the 'Meas Value Operator' domain. Accepted Concepts. Leave it NULL if there's an	integer	No	No	Yes	CONCEPT	

	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 =.	exact numeric value given (instead of putting ‘=’) or there’s no numeric value at all.						
value_as_number	This is the numerical value of the Result of the Measurement, if available. Note that measurements such as blood pressures will be split into their component parts i.e. one record for systolic, one record for diastolic.	Convention for negative values	float	No	No	No		
value_as_concept_id	If the raw data gives a categorial result for measurements those values are captured and mapped to standard concepts in the	If there is no categorial result in the source data, set VALUE_AS_CONCEPT_ID to NULL, if there is a categorial result in a source data but	integer	No	No	Yes	CONCEPT	

	‘Meas Value’ domain.	without mapping, set VALUE_AS_CONCEPT_ID to 0, else map to a CONCEPT_ID.						
unit_concept_id	At present, there isn’t a prescribed unit for individual measurements, such as Hemoglobin A1C, meaning it’s not obligatory to express these measurements as a percentage. UNIT_SOURCE_VALUES should be linked to a Standard Concept within the Unit domain that most accurately reflects the unit provided in the source data.	If the source data does not include units, set UNIT_CONCEPT_ID to NULL. If units are provided but not mapped, set UNIT_CONCEPT_ID to 0. Otherwise, map the units to a CONCEPT_ID. Remember that units are case-sensitive in vocabulary.	integer	No	No	Yes	CONCEPT	Unit
range_low	Ranges have the same unit as the VALUE_AS_NUMBER. These ranges are provided by the source and	If reference ranges for upper and lower limit of normal as provided (typically by a laboratory) these are stored in the	float	No	No	No		

	should remain NULL if not given.	RANGE_HIGH and RANGE_LOW fields. This should be set to NULL if not provided.					
range_high	Ranges have the same unit as the VALUE_AS_NUMBER. These ranges are provided by the source and should remain NULL if not given.	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. This should be set to NULL if not provided.	float	No	No	No	
provider_id	The provider associated with measurement record, e.g. the provider who ordered the test or the provider who recorded the result.	The ETL may need to make a choice as to which PROVIDER_ID to put here. Based on what is available this may or may not be different than the provider associated with the overall VISIT_OCCURRENCE record. For example the admitting vs attending	integer	No	No	Yes	PROVIDER

		physician on an EHR record.						
visit_occurrence_id	The visit during which the Measurement occurred.	Depending on the structure of the source data, this may have to be determined based on dates. If a MEASUREMENT_DATE occurs within the start and end date of a Visit it is a valid ETL choice to choose the VISIT_OCCURRENCE_ID from the visit that subsumes it, even if not explicitly stated in the data. While not required, an attempt should be made to locate the VISIT_OCCURRENCE_ID of the measurement record. If a measurement is related to a visit explicitly in the source data, it is possible that the result date of the Measurement falls outside of the	integer	No	No	Yes	VISIT_OCCURRENCE	

		bounds of the Visit dates.					
visit_detail_id	The VISIT_DETAIL record during which the Measurement occurred. For example, if the Person was in the ICU at the time the VISIT_OCCURRENCE record would reflect the overall hospital stay and the VISIT_DETAIL record would reflect the ICU stay during the hospital visit.	Same rules apply as for the VISIT_OCCURRENCE_ID.	integer	No	No	Yes	VISIT_DETAIL
measurement_source_value	This field contains the exact value from the source data that represents the measurement that occurred.	This value corresponds to a standardized CONCEPT_ID found in MEASUREMENT_CONCEPT_ID and in the 'Measurement' domain within the Standardized Vocabularies. The original code is	varchar(50)	No	No	No	

		retained here for reference purposes.					
measurement_source_concept_id	This is the concept representing the MEASUREMENT_SOURCE_VALUE and may not necessarily be standard. This field is discouraged from use in analysis because it is not required to contain Standard Concepts that are used across the OHDSI community, and should only be used when Standard Concepts do not adequately represent the source detail for the Measurement necessary for a given analytic use case. Consider using MEASUREMENT_CONCEPT_ID instead to enable standardized	If the MEASUREMENT_SOURCE_VALUE is coded in the source data using a vocabulary supported by OMOP Standardized Vocabularies, insert the CONCEPT_ID representing the source value here.	integer	No	No	Yes	CONCEPT

	analytics that can be consistent across the network.						
unit_source_value	This field contains the exact value from the source data that represents the unit of measurement used.	This value corresponds to a standardized CONCEPT_ID found in UNIT_CONCEPT_ID and in the 'Unit' domain within the Standardized Vocabularies. The original code is retained here for reference purposes.	varchar(50)	No	No	No	
unit_source_concept_id	"This is the concept representing the UNIT_SOURCE_VALUE and may not necessarily be standard. This field is discouraged from use in analysis because it is not required to contain Standard Concepts that are used across the OHDSI community, and	If the UNIT_SOURCE_VALUE is coded in the source data using an OMOP supported vocabulary put the concept id representing the source value here.	integer	No	No	Yes	CONCEPT

	should only be used when Standard Concepts do not adequately represent the source detail for the Measurement necessary for a given analytic use case. Consider using UNIT_CONCEPT_ID instead to enable standardized analytics that can be consistent across the network.”					
value_source_value	This field houses the verbatim result value of the Measurement from the source data .	If both a continuous and categorical result are given in the source data such that both VALUE_AS_NUMBER and VALUE_AS_CONCEPT_ID are both included, store the verbatim value that was mapped to VALUE_AS_CONCEPT_ID here.	varchar(50)	No	No	No

measurement_event_id	If the Measurement record is related to another record in the database, this field is the primary key of the linked record.	Put the primary key of the linked record, if applicable, here.	integer	No	No	No	
meas_event_field_concept_id	If the Measurement record is related to another record in the database, this field is the CONCEPT_ID that identifies which table the primary key of the linked record came from.	Put the CONCEPT_ID that identifies which table and field the MEASUREMENT_EVENT_ID came from.	integer	No	No	Yes	CONCEPT

observation

Table Description

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.

User Guide

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 but they must not belong to the Condition, Procedure, Drug, Device, Specimen, or Measurement domains and they must be Standard Concepts.

The observation table usually records the date or datetime of when the observation was obtained, not the date of the observation starting. For example, if the patient reports that they had a heart attack when they were 50, the observation date or datetime is the date of the report, the heart attack observation can have a value_as_concept which captures how long ago the observation applied to the patient.

ETL Conventions

Records whose Source Values map to any domain besides Condition, Procedure, Drug, Specimen, Measurement or Device should be stored in the Observation table. Observations can be stored as attribute value pairs, with the attribute as the Observation Concept and the value representing the clinical fact. This fact can be a Concept (stored in VALUE_AS_CONCEPT), a numerical value (VALUE_AS_NUMBER), a verbatim string (VALUE_AS_STRING), or a datetime (VALUE_AS_DATETIME). Even though Observations do not have an explicit result, the clinical fact can be stated separately from the type of Observation in the VALUE_AS_* fields. 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.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
observation_id	The unique key given to an Observation record for a Person. Refer to the ETL for how duplicate Observations during the same	Each instance of an observation present in the source data should be assigned this unique key.	integer	Yes	Yes	No		

	Visit were handled.							
person_id	The PERSON_ID of the Person for whom the Observation is recorded. This may be a system generated code.		integer	Yes	No	Yes	PERSON	
observation_concept_id	The OBSERVATION_CONCEPT_ID field is recommended for primary use in analyses, and must be used for network studies.	The CONCEPT_ID that the OBSERVATION_SOURCE_CONCEPT_ID maps to. There is no specified domain that the Concepts in this table must adhere to. The only rule is that records with Concepts in the Condition, Procedure, Drug, Measurement, or Device domains MUST go to the corresponding table.	integer	Yes	No	Yes	CONCEPT	
observation_date	The date of when the Observation was obtained. Depending on what the Observation represents this	For some observations the ETL may need to make a choice as to which date to choose.	date	Yes	No	No		

	could be the date of a lab test, the date of a survey, or the date a patient’s family history was taken.							
observation_datetime		If no time is given set to midnight (00:00:00).	datetime	No	No	No		
observation_type_concept_id	This field can be used to determine the provenance of the Observation record, as in whether the measurement was from an EHR system, insurance claim, registry, or other sources.	Choose the OBSERVATION_TYPE_CONCEPT_ID that best represents the provenance of the record, for example whether it came from an EHR record or billing claim. Accepted Concepts. A more detailed explanation of each Type Concept can be found on the vocabulary wiki.	integer	Yes	No	Yes	CONCEPT	Type Concept
value_as_number	This is the numerical value of the Result of the Observation, if applicable and available. It is not expected that all Observations will		float	No	No	No		

	have numeric results, rather, this field is here to house values should they exist.							
value_as_string	This is the categorical value of the Result of the Observation, if applicable and available.		varchar(60)	No	No	No		
value_as_concept_id	It is possible that some records destined for the Observation table have two clinical ideas represented in one source code. This is common with ICD10 codes that describe a family history of some Condition, for example. In OMOP the Vocabulary breaks these two clinical ideas into two codes; one becomes the OBSERVATION_CONCEPT_ID and the other	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, ICD10 Z82.4 'Family history of ischaemic heart disease and other diseases of the circulatory system'	Integer	No	No	Yes	CONCEPT	

	becomes the VALUE_AS_CONCEPT_ID. It is important when using the Observation table to keep this possibility in mind and to examine the VALUE_AS_CONCEPT_ID field for relevant information.	has a ‘Maps to’ relationship to 4167217 ‘Family history of clinical finding’ as well as a ‘Maps to value’ record to 134057 ‘Disorder of cardiovascular system’. If there’s no categorial result in a source_data, set value_as_concept_id to NULL, if there is a categorial result in a source_data but without mapping, set value_as_concept_id to 0.					
qualifier_concept_id	This field contains all attributes specifying the clinical fact further, such as as degrees, severities, drug-drug interaction alerts etc.	Use your best judgement as to what Concepts to use here and if they are necessary to accurately represent the clinical record. There is no restriction on the domain of these Concepts, they just need to be Standard.	integer	No	No	Yes	CONCEPT

unit_concept_id	There is currently no recommended unit for individual observation concepts. UNIT_SOURCE_VALUES should be mapped to a Standard Concept in the Unit domain that best represents the unit as given in the source data.	There is no standardization requirement for units associated with OBSERVATION_CONCEPT_IDs, however, it is the responsibility of the ETL to choose the most plausible unit. If the source unit is NULL (applicable to cases when there’s no numerical value or when it doesn’t require a unit), keep unit_concept_id NULL as well. If there’s no mapping of a source unit, populate unit_concept_id with 0.	integer	No	No	Yes	CONCEPT	Unit
provider_id	The provider associated with the observation record, e.g. the provider who ordered the test or the provider who recorded the result.	The ETL may need to make a choice as to which PROVIDER_ID to put here. Based on what is available this may or may not be different than the provider associated with the overall VISIT_OCCURRENCE record. For example the admitting vs	integer	No	No	Yes	PROVIDER	

		attending physician on an EHR record.					
visit_occurrence_id	The visit during which the Observation occurred.	Depending on the structure of the source data, this may have to be determined based on dates. If an OBSERVATION_DATE occurs within the start and end date of a Visit it is a valid ETL choice to choose the VISIT_OCCURRENCE_ID from the visit that subsumes it, even if not explicitly stated in the data. While not required, an attempt should be made to locate the VISIT_OCCURRENCE_ID of the observation record. If an observation is related to a visit explicitly in the source data, it is possible that the result date of the Observation falls outside of the bounds of the Visit dates.	integer	No	No	Yes	VISIT_OCCURRENCE

visit_detail_id	The VISIT_DETAIL record during which the Observation occurred. For example, if the Person was in the ICU at the time the VISIT_OCCURRENCE record would reflect the overall hospital stay and the VISIT_DETAIL record would reflect the ICU stay during the hospital visit.	Same rules apply as for the VISIT_OCCURRENCE_ID.	integer	No	No	Yes	VISIT_DETAIL
observation_source_value	This field houses the verbatim value from the source data representing the Observation that occurred. For example, this could be an ICD10 or Read code.	This code is mapped to a Standard Concept in the Standardized Vocabularies and the original code is stored here for reference.	varchar(50)	No	No	No	

observation_source_concept_id	This is the concept representing the OBSERVATION_SOURCE_VALUE and may not necessarily be standard. This field is discouraged from use in analysis because it is not required to contain Standard Concepts that are used across the OHDSI community, and should only be used when Standard Concepts do not adequately represent the source detail for the Observation necessary for a given analytic use case. Consider using OBSERVATION_CONCEPT_ID instead to enable standardized analytics that can be consistent	If the OBSERVATION_SOURCE_VALUE is coded in the source data using an OMOP supported vocabulary put the concept id representing the source value here.	integer	No	No	Yes	CONCEPT	

	across the network.					
unit_source_value	This field houses the verbatim value from the source data representing the unit of the Observation that occurred.	This code is mapped to a Standard Condition Concept in the Standardized Vocabularies and the original code is stored here for reference.	varchar(50)	No	No	No
qualifier_source_value	This field houses the verbatim value from the source data representing the qualifier of the Observation that occurred.	This code is mapped to a Standard Condition Concept in the Standardized Vocabularies and the original code is stored here for reference.	varchar(50)	No	No	No
value_source_value	This field houses the verbatim result value of the Observation from the source data. Do not get confused with the Observation_source_value which captures source value of the observation mapped to observation_concept_id. This	If the observation_source_value was a question, for example, or an observation that requires a result then this field is the answer/ result from the source data. Store the verbatim value that represents the result of the observation_source_value.	varchar(50)	No	No	No

	field is the observation result value from the source.						
observation_event_id	If the Observation record is related to another record in the database, this field is the primary key of the linked record.	Put the primary key of the linked record, if applicable, here. See the ETL Conventions for the OBSERVATION table for more details.	integer	No	No	No	
obs_event_field_concept_id	If the Observation record is related to another record in the database, this field is the CONCEPT_ID that identifies which table the primary key of the linked record came from.	Put the CONCEPT_ID that identifies which table and field the OBSERVATION_EVENT_ID came from.	integer	No	No	Yes	CONCEPT

death

Table Description

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 in an administrative claim, status of enrollment into a health plan, or explicit record in EHR data.

User Guide

NA

ETL Conventions

For specific conventions on how to populate this table, please refer to the THEMIS repository.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
person_id			integer	Yes	No	Yes	PERSON	
death_date	The date the person was deceased.	If the precise date include 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. For additional conventions related to this field, please refer to the THEMIS repository.	date	Yes	No	No		

death_datetime		If you have date and time of death, populate death_datetime, otherwise leave NULL	datetime	No	No	No		
death_type_concept_id	This is the provenance of the death record, i.e., where it came from. It is possible that an administrative claims database would source death information from a government file so do not assume the Death Type is the same as the Visit Type, etc.	Use the type concept that reflects the source of the death record. Accepted Concepts. A more detailed explanation of each Type Concept can be found on the vocabulary wiki.	integer	No	No	Yes	CONCEPT	Type Concept

cause_concept_id	This is the Standard Concept representing the Person’s cause of death, if available.	There is no specified domain for this concept, just choose the Standard Concept Id that best represents the person’s cause of death.	integer	No	No	Yes	CONCEPT
cause_source_value		If available, put the source code representing the cause of death here.	varchar(50)	No	No	No	
cause_source_concept_id		If the cause of death was coded using a Vocabulary present in the OMOP Vocabularies (not necessarily a standard concept) put the CONCEPT_ID representing the cause of death here.	integer	No	No	Yes	CONCEPT

note

Table Description

The NOTE table captures unstructured information that was recorded by a provider about a patient in free text (in ASCII, or preferably in UTF8 format) notes on a given date. The type of note_text is CLOB or varchar(MAX) depending on RDBMS.

User Guide

NA

ETL Conventions

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 CMS'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 rolls up to a unique LOINC code.

According to CDO requirements, only 2 of the 5 dimensions are required to properly annotate a document; Kind of Document and any one of the other 4 dimensions. However, not all the permutations of the CDO dimensions will necessarily yield an existing LOINC code. Each of these dimensions are contained in the OMOP Vocabulary under the domain of 'Meas Value' with each dimension represented as a Concept Class.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain

note_id	A unique identifier for each note.		integer	Yes	Yes	No		
person_id			integer	Yes	No	Yes	PERSON	
note_date	The date the note was recorded.		date	Yes	No	No		
note_datetime		If time is not given set the time to midnight.	datetime	No	No	No		
note_type_concept_id	The provenance of the note. Most likely this will be EHR.	Put the source system of the note, as in EHR record. Accepted Concepts. A more detailed explanation of each Type Concept can be found on the vocabulary wiki.	integer	Yes	No	Yes	CONCEPT	Type Concept

note_class_concept_id	A Standard Concept Id representing the HL7 LOINC Document Type Vocabulary classification of the note.	Map the note classification to a Standard Concept. For more information see the ETL Conventions in the description of the NOTE table. [Accepted Concepts](). This Concept can alternatively be represented by concepts with the relationship 'Kind of (LOINC)' to [706391]() (Note).	integer	Yes	No	Yes	CONCEPT
note_title	The title of the note.		varchar(250)	No	No	No	
note_text	The content of the note.		varchar(MAX)	Yes	No	No	
encoding_concept_id	This is the Concept representing the character	Put the Concept Id that represents the encoding character type here. Currently the only option is UTF-8 ([32678]()). It the note is encoded in any	integer	Yes	No	Yes	CONCEPT

	encoding type.	other type, like ASCII then put 0.					
language_concept_id	The language of the note.	Use Concepts that are descendants of the concept 4182347 (World Languages).	integer	Yes	No	Yes	CONCEPT
provider_id	The Provider who wrote the note.	The ETL may need to make a determination on which provider to put here.	integer	No	No	Yes	PROVIDER
visit_occurrence_id	The Visit during which the note was written.		integer	No	No	Yes	VISIT_OCCURRENCE
visit_detail_id	The Visit Detail during which the note was written.		integer	No	No	Yes	VISIT_DETAIL

note_source_value		The source value mapped to the NOTE_CLASS_CONCEPT_ID.	varchar(50)	No	No	No
note_event_id	If the Note record is related to another record in the database, this field is the primary key of the linked record.	Put the primary key of the linked record, if applicable, here.	integer	No	No	No

note_event_field_concept_id	If the Note record is related to another record in the database, this field is the CONCEPT_ID that identifies which table the primary key of the linked record came from.	Put the CONCEPT_ID that identifies which table and field the NOTE_EVENT_ID came from.	integer	No	No	Yes	CONCEPT

note_nlp

Table Description

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

User Guide

NA

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
note_nlp_id	A unique identifier for the NLP record.		integer	Yes	Yes	No		
note_id	This is the NOTE_ID for the NOTE record the NLP record is associated to.		integer	Yes	No	No		

section_concept_id		The SECTION_CONCEPT_ID should be used to represent the note section contained in the NOTE_NLP record. These concepts can be found as parts of document panels and are based on the type of note written, i.e. a discharge summary. These panels can be found as concepts with the relationship ‘Subsumes’ to CONCEPT_ID 45875957.	integer	No	No	Yes	CONCEPT
snippet	A small window of text surrounding the term		varchar(250)	No	No	No	
“offset”	Character offset of the extracted term in		varchar(50)	No	No	No	

	the input note						
lexical_variant	Raw text extracted from the NLP tool.		varchar(250)	Yes	No	No	
note_nlp_concept_id			integer	No	No	Yes	CONCEPT
note_nlp_source_concept_id			integer	No	No	Yes	CONCEPT
nlp_system		Name and version of the NLP system that extracted the term. Useful for data provenance.	varchar(250)	No	No	No	
nlp_date	The date of the note processing.		date	Yes	No	No	
nlp_datetime	The date and time of the note processing.		datetime	No	No	No	

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/li> Rule_out = true Uncertain = very low certainty or any lower certainties A complete lack of modifiers would make Term_exists true.	varchar(1)	No	No	No
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 =	varchar(50)	No	No	No

	anything before the time of the report				
term_modifiers	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_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.	varchar(2000)	No	No	No

specimen

Table Description

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

User Guide

NA

ETL 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.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
specimen_id	Unique identifier for each specimen.		integer	Yes	Yes	No		
person_id	The person from whom the specimen is collected.		integer	Yes	No	Yes	PERSON	
specimen_concept_id		The standard CONCEPT_ID that the SPECIMEN_SOURCE_VALUE maps to in the	integer	Yes	No	Yes	CONCEPT	

		specimen domain. Accepted Concepts						
specimen_type_concept_id		Put the source of the specimen record, as in an EHR system. Accepted Concepts. A more detailed explanation of each Type Concept can be found on the vocabulary wiki.	integer	Yes	No	Yes	CONCEPT	Type Concept
specimen_date	The date the specimen was collected.		date	Yes	No	No		
specimen_datetime			datetime	No	No	No		
quantity	The amount of specimen collected from the person.		float	No	No	No		

unit_concept_id	The unit for the quantity of the specimen.	Map the UNIT_SOURCE_VALUE to a Standard Concept in the Unit domain. Accepted Concepts. If the source unit is NULL (applicable to cases when there's no numerical value or when it doesn't require a unit), keep unit_concept_id NULL as well. If there's no mapping of a source unit, populate unit_concept_id with 0.	integer	No	No	Yes	CONCEPT
anatomic_site_concept_id	This is the site on the body where the specimen is from.	Map the ANATOMIC_SITE_SOURCE_VALUE to a Standard Concept in the Spec Anatomic Site domain. This should be coded at the lowest level of granularity Accepted Concepts	integer	No	No	Yes	CONCEPT
disease_status_concept_id			integer	No	No	Yes	CONCEPT
specimen_source_id	This is the identifier for the speci		varchar (50)	No	No	No	

	men from the source system.				
specimen_source_value		varchar (50)	No	No	No
unit_source_value	This unit for the quantity of the specimen, as represented in the source.	varchar (50)	No	No	No
anatomic_site_source_value	This is the site on the body where the specimen was taken from, as represented in the source.	varchar (50)	No	No	No
disease_status_source_value		varchar (50)	No	No	No

fact_relationship

Table Description

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

User Guide

NA

ETL 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

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
domain_concept_id_1			integer	Yes	No	Yes	CONCEPT	
fact_id_1			integer	Yes	No	No		
domain_concept_id_2			integer	Yes	No	Yes	CONCEPT	
fact_id_2			integer	Yes	No	No		
relationship_concept_id			integer	Yes	No	Yes	CONCEPT	

location

Table Description

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

User Guide

The current iteration of the LOCATION table is US centric. Until a major release to correct this, certain fields can be used to represent different international values.

- STATE can also be used for province or district
- ZIP is also the postal code or postcode
- COUNTY can also be used to represent region

ETL 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.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
location_id	The unique key given to a unique Location.	Each instance of a Location in the source data should be assigned this unique key.	integer	Yes	Yes	No		
address_1	This is the first line of the address.		varchar(50)	No	No	No		

address_2	This is the second line of the address	varchar(50)	No	No	No
city		varchar(50)	No	No	No
state		varchar(2)	No	No	No

zip	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	varchar(9)	No	No	No

		digits. For international addresses, different rules apply.					
county			varchar(20)	No	No	No	
location_source_value		Put the verbatim value for the location here, as it shows up in the source.	varchar(50)	No	No	No	
country_concept_id	The Concept Id representing the country. Values should conform to the Geography domain.		integer	No	No	Yes	CONCEPT
country_source_value	The name of the country.		varchar(80)	No	No	No	
latitude		Must be between	float	No	No	No	

		-90 and 90.				
longitude		Must be between -180 and 180.	float	No	No	No

care_site

Table Description

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.).

User Guide

NA

ETL Conventions

Care site is a unique combination of location_id and nature of the site - the latter could be the place of service, name, or another characteristic in your source data. 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 can be 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.

For additional detailed conventions on how to populate this table, please refer to THEMIS repository.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
care_site_id		Assign an ID to each combination of a location and nature of the site - the latter could be the Place of Service, name or another characteristic in your source data.	integer	Yes	Yes	No		
care_site_name	The name of the care_site as it appears in the source data		varchar(255)	No	No	No		

place_of_service_concept_id	This is a high-level way of characterizing a Care Site. Typically, however, Care Sites can provide care in multiple settings (inpatient, outpatient, etc.) and this granularity should be reflected in the visit.	Choose the concept in the visit domain that best represents the setting in which healthcare is provided in the Care Site. If most visits in a Care Site are Inpatient, then the place_of_service_concept_id should represent Inpatient. If information is present about a unique Care Site (e.g. Pharmacy) then a Care Site record should be created. Accepted Concepts. For information about how to populate this field please see the THEMIS Conventions.	integer	No	No	Yes	CONCEPT
location_id	The location_id from the LOCATION table representing the physical location of the		integer	No	No	Yes	LOCATION

	care_site.					
care_site_source_value	The identifier of the care_site as it appears in the source data. This could be an identifier separate from the name of the care_site.		varchar(50)	No	No	No
place_of_service_source_value		Put the place of service of the care_site as it appears in the source data.	varchar(50)	No	No	No

provider

Table Description

The PROVIDER table contains a list of uniquely identified healthcare providers; duplication is not allowed. These are individuals providing hands-on healthcare to patients, such as physicians, nurses, midwives, physical therapists etc.

User Guide

Many sources do not make a distinction between individual and institutional providers. The PROVIDER table contains the individual providers. If the source only provides limited information such as specialty instead of uniquely identifying individual providers, generic or 'pooled' Provider records are listed in the PROVIDER table.

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
provider_id	It is assumed that every provider with a different unique identifier is in fact a different person and should be treated independently.	This identifier can be the original id from the source data provided it is an integer, otherwise it can be an autogenerated number.	integer	Yes	Yes	No		
provider_name	This field contains information that describes a healthcare provider.	This field is not required for identifying the Provider's actual identity. Instead, its purpose is to uniquely	varchar(255)	No	No	No		

		and/or anonymously identify providers of care across the database.				
npi	This is the National Provider Number issued to health care providers in the US by the Centers for Medicare and Medicaid Services (CMS).		varchar(20)	No	No	No
dea	This is the identifier issued by the DEA, a US federal agency, that allows a provider to write prescriptions for controlled		varchar(20)	No	No	No

	substances.							
specialty_concept_id	This field either represents the most common specialty that occurs in the data or the most specific concept that represents all specialties listed, should the provider have more than one. This includes physician specialties such as internal medicine, emergency medicine, etc. and allied health professionals such as nurses,	If a Provider has more than one Specialty, there are two options: 1. Choose a concept_id which is a common ancestor to the multiple specialties, or, 2. Choose the specialty that occurs most often for the provider. Concepts in this field should be Standard with a domain of Provider. Accepted Concepts.	integer	No	No	Yes	CONCEPT	

	midwives, and pharmacists.							
care_site_id	This is the CARE_SITE_ID for the location that the provider primarily practices in.	If a Provider has more than one Care Site, the main or most often exerted CARE_SITE_ID should be recorded.	integer	No	No	Yes	CARE_SITE	
year_of_birth			integer	No	No	No		
gender_concept_id	This field represents the recorded gender of the provider in the source data.	If given, put a concept from the gender domain representing the recorded gender of the provider. [Accepted Concepts]().	integer	No	No	Yes	CONCEPT	Gender

provider_source_value	Use this field to link back to providers in the source data. This is typically used for error checking of ETL logic.	Some use cases require the ability to link back to providers in the source data. This field allows for the storing of the provider identifier as it appears in the source.	varchar(50)	No	No	No

specialty_source_value	This refers to the specific type of healthcare provider or field of expertise listed in the source data, encompassing physician specialties like internal medicine, emergency medicine, etc., as well as allied health professionals such as nurses, midwives, and pharmacists. It covers medical specialties like surgery, internal medicine, and	The type of provider and their specialty should be entered as they appear in the source data. The decision to use either the coded value or the text description is left to the discretion of the ETL-er.	varchar(50)	No	No	No

	radiology, while other services like prosthetics, acupuncture, and physical therapy fall under the domain of “Service.”						
specialty_source_concept_id	This is often zero as many sites use proprietary codes to store physician speciality.	If the source data codes provider specialty in an OMOP supported vocabulary store the concept_id here.	integer	No	No	Yes	CONCEPT
gender_source_value	This is provider’s gender as it appears in the source data.	Put the provider’s gender as it appears in the source data. This field is up to the discretion of the ETL-er as to whether this should be the coded	varchar(50)	No	No	No	

		value from the source or the text description of the lookup value.					
gender_source_concept_id	This is often zero as many sites use proprietary codes to store provider gender.	If the source data codes provider gender in an OMOP supported vocabulary store the concept_id here.	integer	No	No	Yes	CONCEPT

payer_plan_period

Table Description

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.

User Guide

A Person can have multiple, overlapping, Payer_Plan_Periods in this table. For example, medical and drug coverage in the US can be represented by two Payer_Plan_Periods. The details of the benefit structure of the Plan is rarely known, the idea is just to identify that the Plans are different.

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
payer_plan_period_id	A unique identifier for each unique combination of a Person, Payer, Plan, and Period of time.		integer	Yes	Yes	No		
person_id	The Person covered by the Plan.	A single Person can have multiple, overlapping, PAYER_PLAN_PERIOD records	integer	Yes	No	Yes	PERSON	
payer_plan_period_start_date	Start date of Plan coverage.		date	Yes	No	No		

payer_plan_period_end_date	End date of Plan coverage.		date	Yes	No	No	
payer_concept_id	This field represents the organization who reimburses the provider which administers care to the Person.	Map the payer directly to a standard CONCEPT_ID with the domain_id of ‘Payer’ (Accepted Concepts). This vocabulary is not exhaustive so if there is a value missing, please see the custom concepts page.	integer	No	No	Yes	CONCEPT
payer_source_value	This is the Payer as it appears in the source data.		varchar(50)	No	No	No	
payer_source_concept_id		If the source data codes the Payer in an OMOP supported vocabulary	integer	No	No	Yes	CONCEPT

		store the concept_id here.					
plan_concept_id	This field represents the specific health benefit Plan the Person is enrolled in.	Map the Plan directly to a standard CONCEPT_ID in the 'Plan' vocabulary (Accepted Concepts). This vocabulary is not exhaustive so if there is a value missing, please see the custom concepts page.	integer	No	No	Yes	CONCEPT
plan_source_value	This is the health benefit Plan of the Person as it appears in the source data.		varchar(50)	No	No	No	
plan_source_concept_id		If the source data codes the Plan in an OMOP	integer	No	No	Yes	CONCEPT

		supported vocabulary store the concept_id here.					
sponsor_concept_id	This field represents the sponsor of the Plan who finances the Plan. This includes self-insured, small group health plan and large group health plan.	Map the sponsor directly to a standard CONCEPT_ID with the domain_id of ‘Sponsor’ (Accepted Concepts). This vocabulary is not exhaustive so if there is a value missing, please see the custom concepts page.	integer	No	No	Yes	CONCEPT
sponsor_source_value	The Plan sponsor as it appears in the source data.		varchar(50)	No	No	No	

sponsor_source_concept_id		If the source data codes the sponsor in an OMOP supported vocabulary store the concept_id here.	integer	No	No	Yes	CONCEPT
family_source_value	The common identifier for all people (often a family) that covered by the same policy.	Often these are the common digits of the enrollment id of the policy members.	varchar(50)	No	No	No	
stop_reason_concept_id	This field represents the reason the Person left the Plan, if known.	Map the stop reason directly to a standard CONCEPT_ID with a domain of ‘Plan Stop Reason’ (Accepted Concepts). If one does not exist visit the Custom Concepts pate	integer	No	No	Yes	CONCEPT

		for more information.					
stop_reason_source_value	The Plan stop reason as it appears in the source data.		varchar(50)	No	No	No	
stop_reason_source_concept_id		If the source data codes the stop reason in an OMOP supported vocabulary store the concept_id here.	integer	No	No	Yes	CONCEPT

cost

Table Description

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

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

User Guide

When dealing with summary costs, the cost of the goods or services the provider provides is often not known directly, but derived from the hospital charges multiplied by an average cost-to-charge ratio.

ETL Conventions

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. 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).

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
cost_id			integer	Yes	Yes	No		
cost_event_id			integer	Yes	No	No		
cost_domain_id			varchar(20)	Yes	No	Yes	DOMAIN	
cost_type_concept_id			integer	Yes	No	Yes	CONCEPT	
currency_concept_id			integer	No	No	Yes	CONCEPT	

total_charge	float	No	No	No	
total_cost	float	No	No	No	
total_paid	float	No	No	No	
paid_by_payer	float	No	No	No	
paid_by_patient	float	No	No	No	
paid_patient_copay	float	No	No	No	
paid_patient_coinsurance	float	No	No	No	
paid_patient_deductible	float	No	No	No	
paid_by_primary	float	No	No	No	
paid_ingredient_cost	float	No	No	No	
paid_dispensing_fee	float	No	No	No	
payer_plan_period_id	integer	No	No	No	
amount_allowed	float	No	No	No	
revenue_code_concept_id	integer	No	No	Yes	CONCEPT

revenue_code_source_value	Revenue codes are a method to charge for a class of procedures and conditions in the U.S. hospital system.	varchar(50)	No	No	No	
drg_concept_id		integer	No	No	Yes	CONCEPT
drg_source_value	Diagnosis Related Groups are US codes used to classify hospital cases into one of approximately 500 groups.	varchar(3)	No	No	No	

drug_era

Table Description

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. Every record in the DRUG_EXPOSURE table should be part of a drug era based on the dates of exposure.

User Guide

NA

ETL Conventions

The SQL script for generating DRUG_ERA records can be found here.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
drug_era_id			integer	Yes	Yes	No		
person_id			integer	Yes	No	Yes	PERSON	
drug_concept_id	The drug_concept_id should conform to the concept class 'ingredient' as the drug_era is an era of time where a person is exposed to a particular drug ingredient.		integer	Yes	No	Yes	CONCEPT	Drug

drug_era_start_date		The Drug Era Start Date is the start date of the first Drug Exposure for a given ingredient, with at least 31 days since the previous exposure.	date	Yes	No	No		

drug_era_end_date	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	date	Yes	No	No

		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.						
drug_exposure_count	The count of grouped DRUG_EXPOSURE records that were included in the DRUG_ERA row		integer	No	No	No		

gap_days	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	integer	No	No	No

above assumption of non-stockpiling.

dose_era

Table Description

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.

User Guide

NA

ETL 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. 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.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
dose_era_id			integer	Yes	Yes	No		
person_id			integer	Yes	No	Yes	PERSON	
drug_concept_id	The Concept Id representing the specific drug ingredient.		integer	Yes	No	Yes	CONCEPT	Drug

unit_concept_id	The Concept Id representing the unit of the specific drug ingredient.		integer	Yes	No	Yes	CONCEPT	Unit
dose_value	The numeric value of the dosage of the drug_ingredient.		float	Yes	No	No		
dose_era_start_date	The date the Person started on the specific dosage, with at least 31 days since any prior exposure.		date	Yes	No	No		
dose_era_end_date		The date the Person was no longer exposed to the dosage of the specific drug ingredient. An era is ended if there are 31 days or more	date	Yes	No	No		

	between dosage records.	

condition_era

Table Description

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 and every Condition Occurrence record should be part of a Condition Era. 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.

User Guide

NA

ETL Conventions

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 SQl Script for generating CONDITION_ERA records can be found here 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.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
condition_era_id			integer	Yes	Yes	No		
person_id			integer	Yes	No	Yes	PERSON	
condition_concept_id	The Concept Id representing the Condition.		integer	Yes	No	Yes	CONCEPT	Condition
condition_era_start_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 with at least 31 days		date	Yes	No	No		

	since any prior record of the same Condition.					
condition_era_end_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.		date	Yes	No	No
condition_occurrence_count	The number of individual Condition Occurrences used to construct the condition era.		integer	No	No	No

episode

Table Description

The EPISODE table aggregates lower-level clinical events (VISIT_OCCURRENCE, DRUG_EXPOSURE, PROCEDURE_OCCURRENCE, DEVICE_EXPOSURE) into a higher-level abstraction representing clinically and analytically relevant disease phases,outcomes and treatments. The EPISODE_EVENT table connects qualifying clinical events (VISIT_OCCURRENCE, DRUG_EXPOSURE, PROCEDURE_OCCURRENCE, DEVICE_EXPOSURE) to the appropriate EPISODE entry. For example cancers including their development over time, their treatment, and final resolution.

User Guide

Valid Episode Concepts belong to the 'Episode' domain. For cancer episodes please see [article], for non-cancer episodes please see [article]. If your source data does not have all episodes that are relevant to the therapeutic area, write only those you can easily derive from the data. It is understood that that table is not currently expected to be comprehensive.

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
episode_id	A unique identifier for each Episode.		integer	Yes	Yes	No		
person_id	The PERSON_ID of the PERSON for whom the episode is recorded.		integer	Yes	No	Yes	PERSON	

episode_concept_id	The EPISODE_CONCEPT_ID represents the kind abstraction related to the disease phase, outcome or treatment.	Choose a concept in the Episode domain that best represents the ongoing disease phase, outcome, or treatment. Please see [article] for cancers and [article] for non-cancers describing how these are defined. Accepted Concepts	integer	Yes	No	Yes	CONCEPT	Episode
episode_start_date	The date when the Episode beings.	Please see [article] for how to define an Episode start date.	date	Yes	No	No		
episode_start_datetime	The date and time when the Episode begins.		datetime	No	No	No		
episode_end_date	The date when the instance of the Episode is considered to have ended.	Please see [article] for how to define an Episode end date.	date	No	No	No		
episode_end_datetime	The date when the instance of the Episode is		datetime	No	No	No		

	considered to have ended.							
episode_parent_id	Use this field to find the Episode that subsumes the given Episode record. This is used in the case that an Episode are nested into each other.	If there are multiple nested levels to how Episodes are represented, the EPISODE_PARENT_ID can be used to record this relationship.	integer	No	No	No		
episode_number	For sequences of episodes, this is used to indicate the order the episodes occurred. For example, lines of treatment could be indicated here.	Please see [article] for the details of how to count episodes.	integer	No	No	No		
episode_object_concept_id	A Standard Concept representing the disease phase, outcome, or other abstraction of which the episode consists. For example, if the EPISODE_CONCEPT_ID is [treatment regimen](#) then the EPISODE_OBJECT_CONCEPT_ID should contain the	Episode entries from the ‘Disease Episode’ concept class should have an episode_object_concept_id that comes from the Condition domain. Episode entries from the ‘Treatment Episode’ concept class should have an	integer	Yes	No	Yes	CONCEPT	Procedure, Regimen

	chemotherapy regimen concept, like Afatinib monotherapy.	episode_object_concept_id that scome from the ‘Procedure’ domain or ‘Regimen’ concept class.						
episode_type_concept_id	This field can be used to determine the provenance of the Episode record, as in whether the episode was from an EHR system, insurance claim, registry, or other sources.	Choose the EPISODE_TYPE_CONCEPT_ID that best represents the provenance of the record. Accepted Concepts. A more detailed explanation of each Type Concept can be found on the vocabulary wiki.	integer	Yes	No	Yes	CONCEPT	Type Concept
episode_source_value	The source code for the Episode 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.		varchar(50)	No	No	No		

episode_source_concept_id	A foreign key to a Episode Concept that refers to the code used in the source.	Given that the Episodes are user-defined it is unlikely that there will be a Source Concept available. If that is the case then set this field to zero.	integer	No	No	Yes	CONCEPT	

episode_event

Table Description

The EPISODE_EVENT table connects qualifying clinical events (such as CONDITION_OCCURRENCE, DRUG_EXPOSURE, PROCEDURE_OCCURRENCE, MEASUREMENT) to the appropriate EPISODE entry. For example, linking the precise location of the metastasis (cancer modifier in MEASUREMENT) to the disease episode.

User Guide

This connecting table is used instead of the FACT_RELATIONSHIP table for linking low-level events to abstracted Episodes.

ETL Conventions

Some episodes may not have links to any underlying clinical events. For such episodes, the EPISODE_EVENT table is not populated.

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
episode_id	Use this field to link the EPISODE_EVENT record to its EPISODE.	Put the EPISODE_ID that subsumes the EPISODE	integer	Yes	No	Yes	EPISODE	

		_EVENT record here.						
event_id	This field is the primary key of the linked record in the database. For example, if the Episode Event is a Condition Occurrence, then the CONDITION_OCCURRENCE_ID of the linked record goes in this field.	Put the primary key of the linked record here.	integer	Yes	No	No		
episode_event_field_concept_id	This field is the CONCEPT_ID that identifies which table the primary key of the linked record came from.	Put the CONCEPT_ID that identifies which table and field the EVENT_ID came from. [Accepted Concepts](#)	integer	Yes	No	Yes	CONCEPT	Metadata

metadata

Table Description

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

User Guide

NA

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
metadata_id	The unique key given to a Metadata record.	Attribute value is auto-generated	integer	Yes	Yes	No		
metadata_concept_id			integer	Yes	No	Yes	CONCEPT	
metadata_type_concept_id			integer	Yes	No	Yes	CONCEPT	
name			varchar(250)	Yes	No	No		
value_as_string			varchar(250)	No	No	No		
value_as_concept_id			integer	No	No	Yes	CONCEPT	

value_as_number	This is the numerical value of the result of the Metadata, if applicable and available. It is not expected that all Metadata will have numeric results, rather, this field is here to house values should they exist.	float	No	No	No
metadata_date		date	No	No	No
metadata_datetime		datetime	No	No	No

cdm_source

Table Description

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

User Guide

NA

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
cdm_source_name	The name of the CDM instance.		varchar(255)	Yes	No	No		
cdm_source_abbreviation	The abbreviation of the CDM instance.		varchar(25)	Yes	No	No		

cdm_holder	The holder of the CDM instance.		varchar(255)	Yes	No	No
source_description	The description of the CDM instance.		varchar(MAX)	No	No	No
source_documentation_reference			varchar(255)	No	No	No
cdm_etl_reference		Version of the ETL script used. e.g. link to the Git release	varchar(255)	No	No	No
source_release_date	The date the data was extracted from the source system. In some systems that is the same as the date the ETL		date	Yes	No	No

	was run. Typically the latest even date in the source is on the source_release_date.						
cdm_release_date	The date the ETL script was completed. Typically this is after the source_release_date.		date	Yes	No	No	
cdm_version	Version of the OMOP CDM used as string. e.g. v5.4		varchar(10)	No	No	No	
cdm_version_concept_id	The Concept Id	You can find all concepts that represent the CDM versions using the query: `SELECT *`	integer	Yes	No	Yes	CONC

	representing the version of the CDM.	`FROM CONCEPT WHERE VOCABULARY_ID = 'CDM' AND CONCEPT_CLASS = 'CDM'`					EPT
vocabulary_version	Version of the OMOP standardised vocabularies loaded	You can find the version of your Vocabulary using the query: `SELECT vocabulary_version from vocabulary where vocabulary_id = 'None'`	varchar(20)	Yes	No	No	

concept

Table Description

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 standardized analytics. Each Standard Concept belongs to one Domain, which defines the location where the Concept would be expected to occur within the data tables of the CDM. Concepts can represent broad categories (‘Cardiovascular disease’), detailed clinical elements (‘Myocardial infarction of the anterolateral wall’), or modifying characteristics and attributes that define Concepts at various levels of detail (severity of a disease, associated morphology, etc.). Records in the Standardized Vocabularies tables are derived from national or international vocabularies such as SNOMED-CT, RxNorm, and LOINC, or custom OMOP Concepts defined to cover various aspects of observational data analysis.

User Guide

The primary purpose of the CONCEPT table is to provide a standardized representation of medical Concepts, allowing for consistent querying and analysis across the healthcare databases. Users can join the CONCEPT table with other tables in the CDM to enrich clinical data with standardized Concept information or use the CONCEPT table as a reference for mapping clinical data from source terminologies to Standard Concepts.

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
concept_id	A unique identifier for each Concept across all domains.		integer	Yes	Yes	No		
concept_name	An unambiguous, meaningful and descriptive name for the Concept.		varchar(255)	Yes	No	No		
domain_id	A foreign key to the DOMAIN table the Concept belongs to.		varchar(20)	Yes	No	Yes	DOMAIN	
vocabulary_id	A foreign key to the VOCABULARY table indicating from which source the Concept has been adapted.		varchar(20)	Yes	No	Yes	VOCABULARY	

concept_class_id	The attribute or concept class of the Concept. Examples are ‘Clinical Drug’, ‘Ingredient’, ‘Clinical Finding’ etc.	varchar(20)	Yes	No	Yes	CONCEPT_CLASS
standard_concept	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.	varchar(1)	No	No	No	
concept_code	The concept code represents the identifier of the Concept in the source vocabulary, such as SNOMED-CT concept IDs, RxNorm RXCUIs	varchar(50)	Yes	No	No	

	etc. Note that concept codes are not unique across vocabularies.							
valid_start_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.		date	Yes	No	No		
valid_end_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.		date	Yes	No	No		

invalid_reason	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.		varchar(1)	No	No	No		

vocabulary

Table Description

The VOCABULARY table includes a list of the Vocabularies integrated from various sources or created de novo in OMOP CDM. This reference table contains a single record for each Vocabulary and includes a descriptive name and other associated attributes for the Vocabulary.

User Guide

The primary purpose of the VOCABULARY table is to provide explicit information about specific vocabulary versions and the references to the sources from which they are asserted. Users can identify the version of a particular vocabulary used in the database, enabling consistency and reproducibility in data analysis. Besides, users can check the vocabulary release version in their CDM which refers to the vocabulary_id = ‘None’.

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain

vocabulary_id	A unique identifier for each Vocabulary, such as ICD9CM, SNOMED, Visit.	varchar(20)	Yes	Yes	No
vocabulary_name	The name describing the vocabulary, for example, International Classification of Diseases, Ninth Revision, Clinical Modification, Volume 1 and 2 (NCHS) etc.	varchar(255)	Yes	No	No
vocabulary_reference	External reference to documentation or available download of the about the vocabulary.	varchar(255)	No	No	No

vocabulary_version	Version of the Vocabulary as indicated in the source.		varchar(255)	No	No	No	
vocabulary_concept_id	A Concept that represents the Vocabulary the VOCABULARY record belongs to.		integer	Yes	No	Yes	CONCEPT

domain

Table Description

The DOMAIN table includes a list of OMOP-defined Domains to which the Concepts of the Standardized Vocabularies can belong. A Domain represents a clinical definition whereby we assign matching Concepts for the standardized fields in the CDM tables. For example, the Condition Domain contains Concepts that describe a patient condition, and these Concepts can only be used 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, including a Domain ID and a descriptive name for every Domain.

User Guide

Users can leverage the DOMAIN table to explore the full spectrum of health-related data Domains available in the Standardized Vocabularies. Also, the information in the DOMAIN table may be used as a reference for mapping source data to OMOP domains, facilitating data harmonization and interoperability.

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
domain_id	A unique key for each domain.		varchar(20)	Yes	Yes	No		
domain_name	The name describing the Domain, e.g. Condition, Procedure, Measurement etc.		varchar(255)	Yes	No	No		
domain_concept_id	A Concept representing the Domain Concept the DOMAIN record belongs to.		integer	Yes	No	Yes	CONCEPT	

concept_class

Table Description

The CONCEPT_CLASS table includes semantic categories that reference the source structure of each Vocabulary. Concept Classes represent so-called horizontal (e.g. MedDRA, RxNorm) or vertical levels (e.g. SNOMED) of the vocabulary structure. Vocabularies without any Concept Classes, such as HCPCS, use the vocabulary_id as the

Concept Class. This reference table is populated with a single record for each Concept Class, which includes a Concept Class ID and a fully specified Concept Class name.

User Guide

Users can utilize the CONCEPT_CLASS table to explore the different classes or categories of concepts within the OHDSI vocabularies.

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
concept_class_id	A unique key for each class.		varchar(20)	Yes	Yes	No		
concept_class_name	The name describing the Concept Class, e.g. Clinical Finding, Ingredient, etc.		varchar(255)	Yes	No	No		
concept_class_concept_id	A Concept that represents the		integer	Yes	No	Yes	CONCEPT	

Concept Class.

concept_relationship

Table Description

The CONCEPT_RELATIONSHIP table contains records that define relationships between any two Concepts and the nature or type of the relationship. This table captures various types of relationships, including hierarchical, associative, and other semantic connections, enabling comprehensive analysis and interpretation of clinical concepts. Every kind of relationship is defined in the RELATIONSHIP table.

User Guide

The CONCEPT_RELATIONSHIP table can be used to explore hierarchical or attribute relationships between concepts to understand the hierarchical structure of clinical concepts and uncover implicit connections and associations within healthcare data. For example, users can utilize mapping relationships (‘Maps to’) to harmonize data from different sources and terminologies, enabling interoperability and data integration across disparate datasets.

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
concept_id_1			integer	Yes	No	Yes	CONCEPT	
concept_id_2			integer	Yes	No	Yes	CONCEPT	

relationship_id	The relationship between CONCEPT_ID_1 and CONCEPT_ID_2. Please see the [Vocabulary Conventions](). for more information.	varchar(20)	Yes	No	Yes	RELATIONSHIP
valid_start_date	The date when the relationship is first recorded.	date	Yes	No	No	
valid_end_date	The date when the relationship is invalidated.	date	Yes	No	No	
invalid_reason	Reason the relationship was invalidated. Possible values are ‘D’ (deleted), ‘U’ (updated) or NULL.	varchar(1)	No	No	No	

relationship

Table Description

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

User Guide

NA

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
relationship_id	The type of relationship captured by the relationship record.		varchar(20)	Yes	Yes	No		
relationship_name			varchar(255)	Yes	No	No		
is_hierarchical	Defines whether a relationship defines concepts into classes or hierarchies. Values are 1 for hierarchical relationship or 0 if not.		varchar(1)	Yes	No	No		

defines_ancestry	Defines whether a hierarchical relationship contributes to the concept_ancestor table. These are subsets of the hierarchical relationships. Valid values are 1 or 0.	varchar(1)	Yes	No	No	
reverse_relationship_id	The identifier for the relationship used to define the reverse relationship between two concepts.	varchar(20)	Yes	No	No	
relationship_concept_id	A foreign key that refers to an identifier in the CONCEPT table for the unique relationship concept.	integer	Yes	No	Yes	CONCEPT

concept_synonym

Table Description

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

User Guide

NA

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
concept_id			integer	Yes	No	Yes	CONCEPT	
concept_synonym_name			varchar(1000)	Yes	No	No		
language_concept_id			integer	Yes	No	Yes	CONCEPT	

concept_ancestor

Table Description

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.

User Guide

NA

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
ancestor_concept_id	The Concept Id for the higher-level concept that forms the ancestor in the relationship.		integer	Yes	No	Yes	CONCEPT	
descendant_concept_id	The Concept Id for the lower-level concept that forms the descendant in the relationship.		integer	Yes	No	Yes	CONCEPT	

min_levels_of_separation	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.	integer	Yes	No	No
max_levels_of_separation	The maximum separation in number of levels of hierarchy between ancestor and descendant concepts. This is an attribute that is used to simplify	integer	Yes	No	No

hierarchic analysis.

source_to_concept_map

Table Description

The source to concept map table is 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. **There are OHDSI tools to help you populate this table; Usagi and Perseus. You can read more about OMOP vocabulary mapping in The Book of OHDSI Chapter 6.3.**

User Guide

NA

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
source_code	The source code being translated into a Standard Concept.		varchar(50)	Yes	No	No		

source_concept_id	A foreign key to the Source Concept that is being translated into a Standard Concept.	This is either 0 or should be a number above 2 billion, which are the Concepts reserved for site-specific codes and mappings.	integer	Yes	No	Yes	CONCEPT
source_vocabulary_id	A foreign key to the VOCABULARY table defining the vocabulary of the source code that is being translated to a Standard Concept.		varchar(20)	Yes	No	No	

source_code_description	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.	varchar(255)	No	No	No	
target_concept_id	The target Concept to which the source code is being mapped.	integer	Yes	No	Yes	CONCEPT
target_vocabulary_id	The Vocabulary of the target Concept.	varchar(20)	Yes	No	Yes	VOCABULARY
valid_start_date	The date when the mapping instance was first recorded.	date	Yes	No	No	

valid_end_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.	date	Yes	No	No
invalid_reason	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.	varchar(1)	No	No	No

drug_strength

Table Description

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.

User Guide

NA

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
drug_concept_id	The Concept representing the Branded Drug or Clinical Drug Product.		integer	Yes	No	Yes	CONCEPT	
ingredient_concept_id	The Concept representing the active ingredient contained within the	Combination Drugs will have more than one record in this table,	integer	Yes	No	Yes	CONCEPT	

	drug product.	one for each active Ingredient.					
amount_value	The numeric value or the amount of active ingredient contained within the drug product.		float	No	No	No	
amount_unit_concept_id	The Concept representing the Unit of measure for the amount of active ingredient contained within the drug product.		integer	No	No	Yes	CONCEPT
numerator_value	The concentration of the active ingredient contained within the		float	No	No	No	

	drug product.					
numerator_unit_concept_id	The Concept representing the Unit of measure for the concentration of active ingredient.	integer	No	No	Yes	CONCEPT
denominator_value	The amount of total liquid (or other divisible product, such as ointment, gel, spray, etc.).	float	No	No	No	
denominator_unit_concept_id	The Concept representing the denominator unit for the concentration of active ingredient.	integer	No	No	Yes	CONCEPT

box_size	The number of units of Clinical Branded Drug or Quantified Clinical or Branded Drug contained in a box as dispensed to the patient.	integer	No	No	No
valid_start_date	The date when the Concept was first recorded. The default value is 1-Jan-1970.	date	Yes	No	No
valid_end_date	The date when then Concept became invalid.	date	Yes	No	No

invalid_reason	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.	varchar(1)	No	No	No

cohort

Table Description

The subject of a cohort can have multiple, discrete records in the cohort table per cohort_definition_id, subject_id, and non-overlapping time periods. The definition of the cohort is contained within the COHORT_DEFINITION table. It is listed as part of the RESULTS schema because it is a table that users of the database as well as tools such as ATLAS need to be able to write to. The CDM and Vocabulary tables are all read-only so it is suggested that the COHORT and COHORT_DEFINTION tables are kept in a separate schema to alleviate confusion.

User Guide

NA

ETL Conventions

Cohorts typically include patients diagnosed with a specific condition, patients exposed to a particular drug, but can also be Providers who have performed a specific Procedure. Cohort records must have a Start Date and an End Date, but the End Date may be set to Start Date or could have an applied censor 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 though they are most often Person Ids. 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

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
cohort_definition_id			integer	Yes	No	No		
subject_id			integer	Yes	No	No		
cohort_start_date			date	Yes	No	No		
cohort_end_date			date	Yes	No	No		

cohort_definition

Table Description

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.

User Guide

NA

ETL Conventions

NA

CDM Field	User Guide	ETL Conventions	Datatype	Required	Primary Key	Foreign Key	FK Table	FK Domain
cohort_definition_id	This is the identifier given to the cohort, usually by the ATLAS application		integer	Yes	No	No		
cohort_definition_name	A short description of the cohort		varchar(255)	Yes	No	No		
cohort_definition_description	A complete description of the cohort.		varchar(MAX)	No	No	No		
definition_type_concept_id	Type defining what kind of Cohort Definition the record represents and how the syntax may be executed.		integer	Yes	No	Yes	CONCEPT	

cohort_definition_syntax	Syntax or code to operationalize the Cohort Definition.		varchar(MAX)	No	No	No		
subject_concept_id	This field contains a Concept that represents the domain of the subjects that are members of the cohort (e.g., Person, Provider, Visit).		integer	Yes	No	Yes	CONCEPT	
cohort_initiation_date	A date to indicate when the Cohort was initiated in the COHORT table.		date	No	No	No		