Implementation Network for Sharing Population Information from Research Entities

(INSPIRE)

ALPHA to OMOP Data and Vocabulary Mapping

Version 1.1

Version 1.1

Version	Date	Author	Reviewer	Description
1.0	2021-01-14	Tathagata Bhattacharjee	Jay Greenfield Chifundo Kanjala	First draft of ALPHA to OMOP data mapping. The process stared in Oct – Nov 2020 with Taurayi Mudzana of SAPRIN and had undergone many revisions before this version
1.1	2021-05-12	Tathagata Bhattacharjee Jay Greenfield	Chifundo Kanjala Maeve Campman Arofan Gregory Jim Todd	Updates incorporated into the document after multiple discussions and reviews done by internal team

ACKNOWLEDGEMENT

This initiative was tasked with mapping the Analysing Longitudinal Population-based HIV/AIDS dada on Africa (ALPHA) residencies and HIV test data specifications to Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM). This project was undertaken as a part of the project funded by the UK Research and Innovation's (UKRI) Global Challenges Research Fund (GCRF) Digital Innovation for Development in Africa (DIDA).

Reference No.: EP/T029315/1

Project Title: Implementation Network for Sharing Population Information from Research Entities in East Africa (INSPIRE-EA)

Table of Contents

ACKNOWLEDGEMENT 3

Multiple drafts 7

Purpose of this draft 7

A Conceptual and Logical Model of Longitudinal Population Research in the Context of OMOP v 6.0 9

Process Outline for ETL Generation 12

OHDSI WhiteRabbit 15

Steps to scan the ALPHA Residencies table using WhiteRabbit 15

OHDSI Rabbit In A Hat 17

Steps to start the mapping process of ALPHA Residencies to OMOP using Rabbit In A Hat 17

OHDSI Usagi 18

Steps to map vocabularies from ALPHA to OMOP using Usagi 18

Mapping ALPHA Residencies to OMOP v6.0 CDM 22

Source Table(s) Structure: 22
Source Data Mapping Approach to CDMV6.0 23
OMOP Table name: person 24
ALPHA Residencies to OMOP Person *24*
Guidelines for Mapping ALPHA Residencies to OMOP Person *24*
SQL Skeleton codes for ETL: ALPHA Residencies to OMOP Person *28*
SQL statement to create the OMOP person table in INSPIRE database's cdm schema. 28
SQL statement to create a Sequence to generate the person_id in PostgreSQL 28
SQL statement to insert data into cdm.person table from alpha.residencies table 29
OMOP Table name: observation_period 30
ALPHA Residencies to OMOP Observation_Period *30*
Guidelines for Mapping ALPHA Residencies to OMOP Observation_Period *30*
SQL Skeleton codes for ETL: ALPHA Residencies to OMOP Observation_Period *31*
SQL statement to create the OMOP observation_period table in INSPIRE database's cdm schema. 31
SQL statement to create a Sequence to generate the observation_period_id in PostgreSQL 31
SQL statement to insert data into cdm.observation_period_id table from alpha.residencies table 31
OMOP Table name: visit_occurrence 32
ALPHA Residencies to OMOP Visit_Occurence *32*
Guidelines for Mapping ALPHA Residencies to OMOP Visit_Occurrence *33*
SQL Skeleton codes for ETL: ALPHA Residencies to OMOP Visit_Occerence *36*
SQL statement to create the OMOP visit_occurence table in INSPIRE database's cdm schema 36
SQL statement to create a Sequence to generate the visit_occurence_id in PostgreSQL 36
SQL statement to insert data into cdm.person table from alpha.residencies table 36
OMOP Table name: observation 38
ALPHA Residencies to OMOP Observation *39*
Guidelines for Mapping ALPHA Residencies to OMOP Observation *39*
SQL Skeleton codes for ETL: ALPHA Residencies to OMOP Observation *44*
SQL statement to create the OMOP observation table in INSPIRE database's cdm schema 44

SQL statement to create a Sequence to generate the observation_id in PostgreSQL 44
SQL statement to insert data into cdm.observation table from alpha.residencies table 44
SQL statement to update the visit_occurrence_id by linking it to the visit_occurrence table 47
OMOP Table name: fact_relationship 47
Guidelines for Mapping ALPHA Residencies to OMOP fact_relationship 48
SQL Skeleton codes for ETL: ALPHA Residencies to OMOP fact_relationship 49
SQL statement to create the OMOP survey_conduct table in INSPIRE database's cdm schema 49
Implementation of ETL to populate fact_relatioship table 49
OMOP Table name: survey_conduct 50
ALPHA Residencies to OMOP Survey_Conduct 50
Guidelines for Mapping ALPHA Residencies to OMOP Survey_Conduct 51
SQL Skeleton codes for ETL: ALPHA Residencies to OMOP Survey_Conduct 54
SQL statement to create the OMOP survey_conduct table in INSPIRE database's cdm schema 54
SQL statement to create a Sequence to generate the survey_conduct_id in PostgreSQL 55
SQL statement to insert data into cdm.survey_conduct table from alpha.residencies table 55
SQL statement to update the visit_occurrence_id by linking it to the visit_occurrence table 58
OMOP Table name: location_history 58
ALPHA Residencies to OMOP Location_History 58
Guidelines for Mapping ALPHA Residencies to OMOP Location_History 58
SQL Skeleton codes for ETL: ALPHA Residencies to OMOP Location_History 59
SQL statement to create the OMOP location_history table in INSPIRE database's cdm schema 60
SQL statement to create a Sequence to generate the visit_location_id in PostgreSQL 60
SQL statement to insert data into cdm.location table from alpha.residencies table 60
Mapping ALPHA HIV Test Data to OMOP v6.0 CDM 61
Source Table(s) Structure: 61
Steps to scan the ALPHA HIV Test table using WhiteRabbit 62
Steps to start the mapping process of ALPHA HIV Test to OMOP using Rabbit In A Hat 65
Source Data Mapping Approach to CDMV6.0 66
OMOP Table name: visit_occurrence 66
ALPHA HIV Test to OMOP visit_occurrence 66
ALPHA HIV Test to OMOP visit_occurrence 67
Guidelines for Mapping ALPHA HIV Test to OMOP Visit_Occurrence 67
SQL Skeleton codes for ETL: ALPHA HIV Test to OMOP Visit_Occurrence 70
SQL statement to create the OMOP visit_occurrence table in INSPIRE database's cdm schema 70
SQL statement to create a Sequence to generate the visit_occurrence_id in PostgreSQL 70
SQL statement to insert data into cdm.visit_occurrence table from alpha.hiv_test table 70
OMOP Table name: condition_occurrence 71
ALPHA HIV Test to OMOP condition_occurrence 71
Guidelines for Mapping ALPHA HIV Test to OMOP Condition_Occurrence 72
SQL Skeleton codes for ETL: ALPHA HIV Test to OMOP Condition_Occurrence 76
SQL statement to create the OMOP condition_occurrence table in INSPIRE database's cdm schema. 76
SQL statement to create a Sequence to generate the condition_occurrence_id in PostgreSQL 76
SQL statement to insert data into cdm.condition_occurrence table from alpha.hiv_test table 76
SQL statement to update the visit_occurrence_id by linking it to the visit_occurrence table 77
Entity Relation Diagram of OMOP CDM Tables after ALPHA Residencies and HIV Tests Data mapping 78
LPS best practices 79

Some demographics domain best practices 79
Some protocol domain best practices 80

A de novo vocabulary for longitudinal population studies 81

Introduction 81
The demographics domain 83
Migration *83*
Other demographic constructs *83*
The protocol domain 83
Periodicity *83*
Temporal relationships *83*
Other protocol constructs *83*

References 85

Multiple drafts

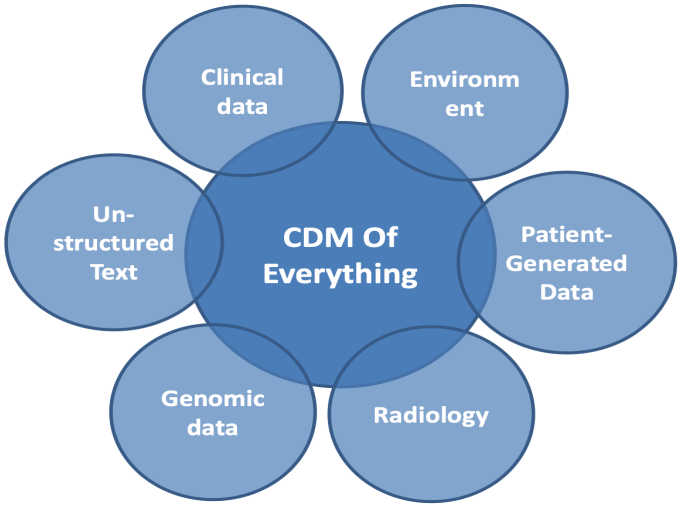
This document will have several drafts. Each draft grows its scope. The scope of the first and the revised draft is described in the next section. It goes to one use of the OMOP CDM platform. The platform promises a grand unification of cohort specific survey and clinical research. In the first draft longitudinal population studies that perform GIS based sentinel surveillance are mapped to the OMOP CDM. In a subsequent draft infectious disease testing and test result reporting are mapped to the OMOP CDM.

Purpose of this draft

The purpose of this document is to first specify a conceptual and logical model for conducting longitudinal population studies (LPS) using the Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM) version 6.0. Then, using this model, this document outlines the intended Extract-Transform-Load (ETL) process for mapping the Analysing Longitudinal Population-based HIV/AIDS data on Africa (ALPHA Network) data specifications, i.e., residencies and HIV testing, to OMOP CDM v 6.0.

This ETL mapping process documentation is built using opensource Observational Health Data Science and Informatics (OHDSI - https://www.ohdsi.org/) tools.

The conceptual and logical model derive, for example, uses of OMOP as a "grand unification across all aspects of health data" including "various types of medical data such as clinical, genomic, radiologic and patient generated data" and both cross-sectional and longitudinal person-specific survey research collected outside of clinical settings. In these example unifications all these data types (aka domains) have been shown to be "indistinguishably accessible in the single [OMOP CDM] database". See Common Data Model of Everything in Medicine for an early effort to describe this "grand unification". Here is its meme:



This document captures the initial process of data mapping between ALPHA data specs and OMOP CDM, which is presumably one of the first attempts globally to map longitudinal population study, the Health and Demographic Surveillance System (HDSS) data into OMOP CDM and thus it is anticipated to have a number of revisions as more experience is gained using OMOP as a “theory of everything medical”.

Here we have attempted to map ALPHA residencies (ALPHA Data Spec 6.1) and HIV Testing (ALPHA Data Spec 6.2b) to OMOP CDM 6.0.

A Conceptual and Logical Model of Longitudinal Population Research in the Context of OMOP v 6.0

The INSPIRE project is developing a method for OMOP CDM to hold and include both cross-sectional and longitudinal population survey (LPS) research. Using OMOP CDM within the context of LPS research is a new development within the field and, therefore, holds potential for growth and innovation through the use of pre-existing tools, allowing us to explore new uses of the OMOP CDM without having to modify the existing set or invent additional "domains" (tables). With the development and incorporation of cross-sectional LPS research, INSPIRE is, in part, working towards more comprehensive unification within health data. These developments work to incorporate health events that both precede and succeed the medical events that transpire in clinical settings through which populations of individuals pass. With this addition of health events and labs recorded in the field, the "CDM of Everything" is positioned to perform 360-degree surveillance of infectious and non-infectious diseases on a single platform. INSPIRE's work positions all LPS research to be gathered or held in OMOP CDM, allowing all applicable data (and metadata) to be commingled and captured in a single instance. INSPIRE supports this with both a conceptual and a logical model.

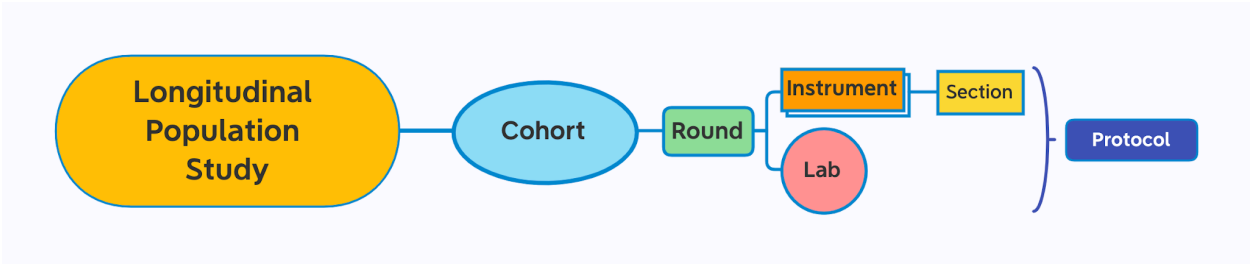


Figure 1: Conceptual Model

This is a generic model that many longitudinal population studies follow. The "Protocol" in this model is a workflow in which one or more cohorts participate. The protocol turns on and off instruments, instrument sections and labs from one round to the next.

In INSPIRE study cohorts and rounds are mapped into data tabulation models aka "specs". There are data tabulation models for demographic surveillance in population studies, HIV surveillance in population studies and COVID-19 surveillance in population studies. Note that INSPIRE also includes data tabulation models aka "specs" for HIV-specific clinical visits (HICDEP 1.120) and COVID-19 clinical visits (WHO/2019-nCoV-Clinical_CRF/2020.4).

The three surveillance data tabulation models adopted by INSPIRE funnel multiple population studies that perform HIV and COVID-19 surveillance undertaken by Health and Demographic Surveillance Systems mostly in low- and middle-income countries (LMIC). Data captured in

these tabulation models is cohort and round specific. These tabulation models are currently in use in many LMIC to facilitate sentinel surveillance.

Using its logical model, INSPIRE is in the process of mapping these surveillance data tabulation models into the OMOP CDM.

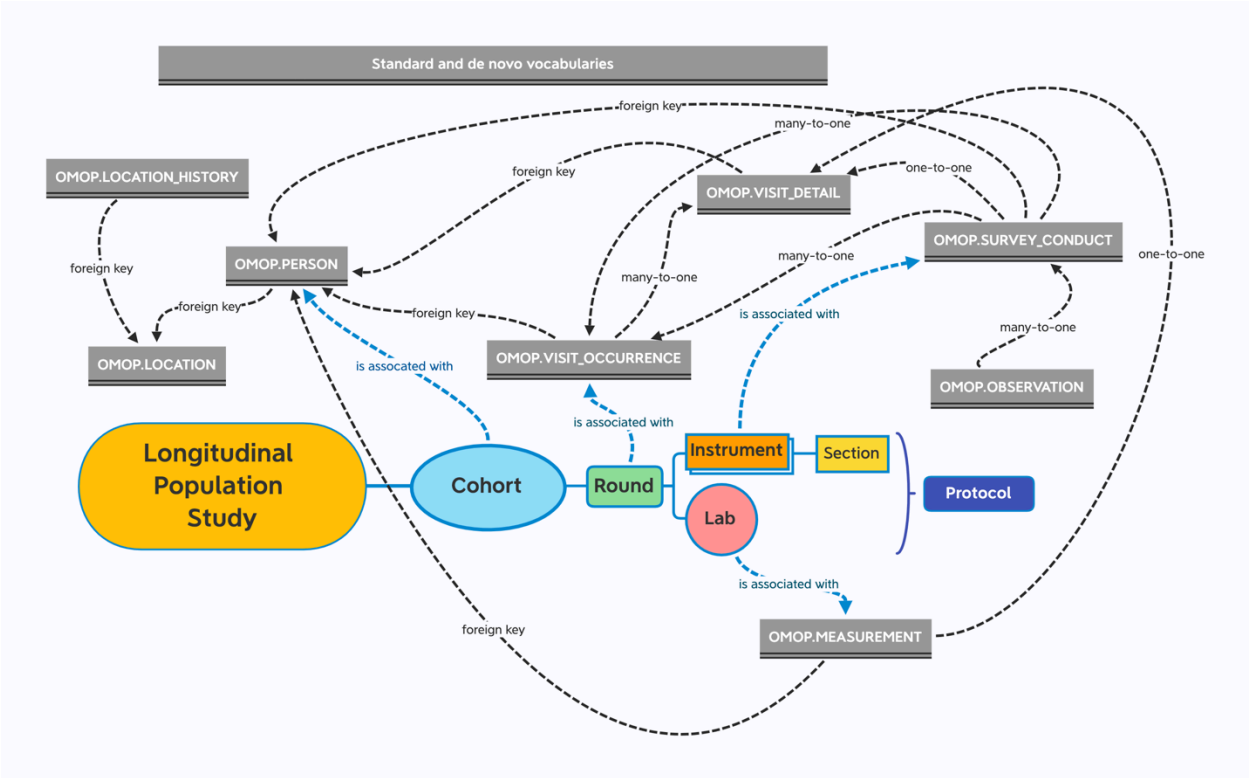


Figure 2: Logical Model

In this logical model we map the LPS conceptual model components as they are represented in the surveillance data tabulation models to various OMOP CDM domains aka tables. Note that a longitudinal population study doesn't use all the OMOP CDM tables. Some of those tables are specific to visits a patient might make to a hospital or clinic.

Surveys that include labs, however, will share certain OMOP CDM tables with clinical visits like the MEASUREMENT table. Also, the VISIT_OCCURRENCE and VISIT_DETAIL OMOP tables are shared by surveillance studies and clinical encounters. In surveillance studies they are mapped to rounds and instruments or labs respectively.

The logical model as presented here, doesn't detail the ***protocol***. The protocol is accounted for by concepts referenced in SURVEY_CONDUCT and VISIT_DETAIL.

Each ***instrument*** in the ***study*** has its own SURVEY_CONDUCT and the SURVEY_CONDUCT table includes a reference to a timing concept in the CONCEPT table. This field is called the **timing_concept_id**. In a subsequent section in this document – A De Novo HDSS Vocabulary – we introduce timing concepts that account for and describe the periodicity of instruments and sections from one ***round*** to the next.

The protocol story continues in VISIT_DETAIL which we have constrained to have a one-to-one relationship with SURVEY_CONDUCT because the **survey_concept_id** and the **visit_detail_concept_id** in an LPS is the same.

In VISIT_DETAIL temporal relations among instruments and ***labs*** **within** a study round are generally accounted for by a link VISIT_DETAIL establishes between successive VISIT_DETAILs by **preceding_detail_visit_id** and/or the **visit_detail_parent_id**. This description can be qualified as needed by a special OBSERVATION. This OBSERVATION has an observation_concept_id that references a temporal relations category set called Allen's interval algebra. Temporal relations among labs, instruments and between labs and instruments are often critical to the success of a VISIT_OCCURRENCE aka round.

Between the **timing_concept_id** in SURVEY_CONDUCT, the ways that VISIT_DETAIL links successive and parent VISIT_DETAILs and the special OBSERVATION that, as needed, can qualify this chain with more granular temporal relationships, we are able to account for the study ***protocol***.

Process Outline for ETL Generation

The sequence of activities followed to generate the ETL process documentation is as follows:

(1) Install WhiteRabbit and Rabbit In A Hat
(2) Scan the source data (data tables) using WhiteRabbit to generate the metadata and fake data
(3) Data mapping documentation using Rabbit In A Hat
(4) Export the data mapping documentation to a MS Word file
(5) Necessary formatting of the MS Word file

Keep in mind that much of the data mapping goes between one or more *wide* tables from the source to two *long* tables in the OMOP CDM target. The *wide* tables host many variables per record. On the other in the OMOP CDM many source variables from these wide tables are mapped either to the OMOP OBSERVATION or the OMOP MEASUREMENT table. Both of these tables take just one variable per record, and, as a consequence, they grow very *long*.

In addition to taking a variable value from their source tables, both OBSERVATION and MEASUREMENT records describe the original concept behind the source variable and a corresponding concept for the same MEASUREMENT or OBSERVATION in the OMOP Standardized Vocabularies.

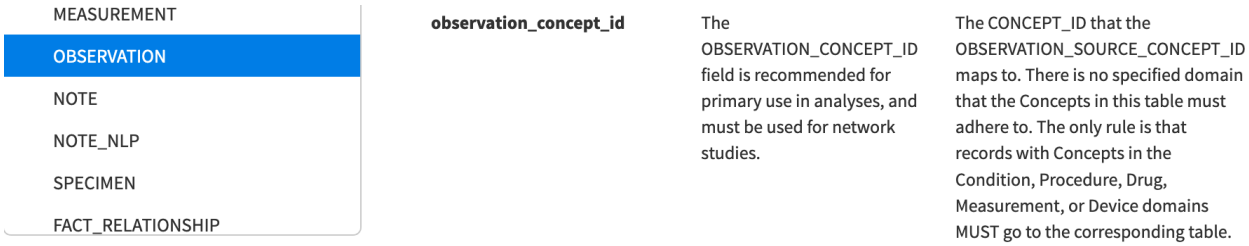


Figure 3: A reference to the concept behind an OBSERVATION in the OMOP Standardized Vocabularies

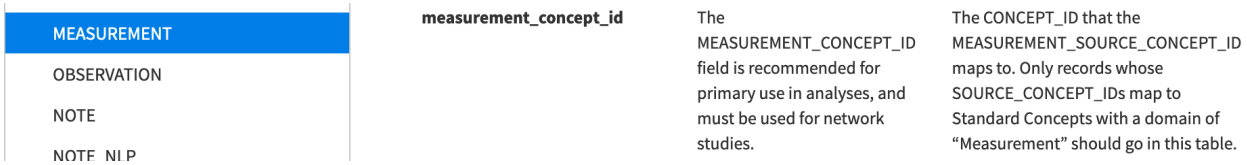


Figure 4: A reference to the concept behind a MEASUREMENT in the OMOP Standardized Vocabularies

Also, when the value of a variable is categorical, this value is captured in either the OBSERVATION or the MESUREMENT record ***not*** as a code but as a concept_id that references the concept behind a category in the Standardized Vocabularies CONCEPT table.

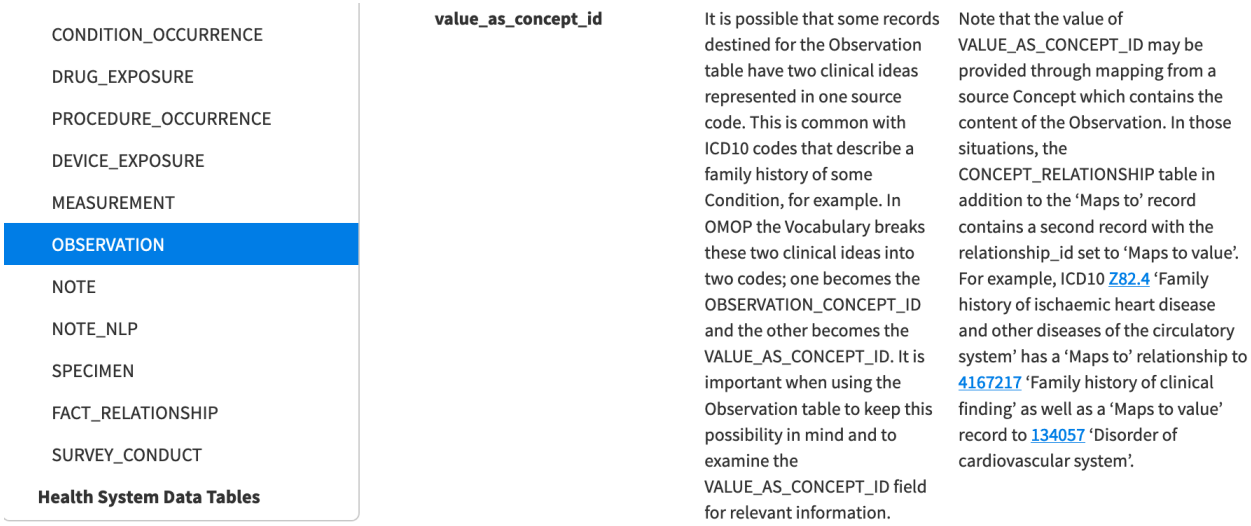


Figure 5: A reference to a CONCEPT for a categorical value taken by an OBSERVATION

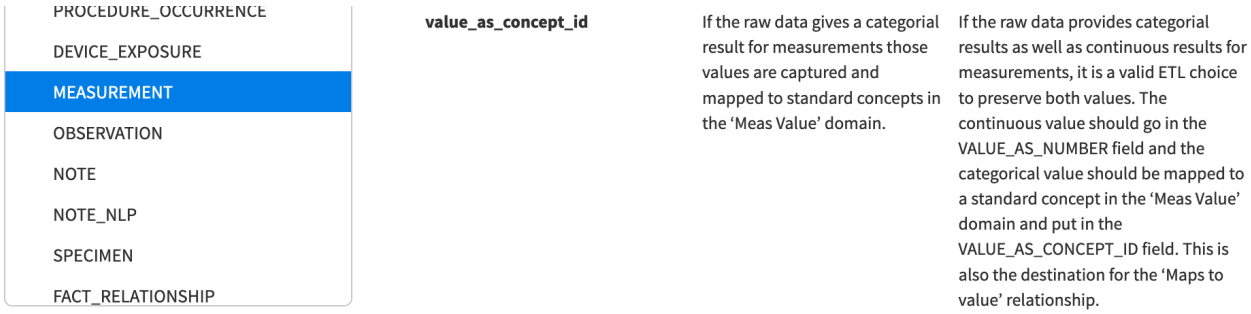


Figure 6: A reference to a CONCEPT for a categorical value taken by a MEASUREMENT

The ETL we build will need to include all of these concept_ids from the OMOP Standardized Vocabularies. In the case of these categorical variables not only are concepts and their concept_ids needed for both a variable and its categories. A CONCEPT_RELATIONSHIP record has to be constructed between this categorical variable and each of its categories. A CONCEPT_RELATIONSHIP record identifies two concept_ids and the relationship they enjoy. The relationship takes the form of a foreign key relationship_id into the RELATIONSHIP table. It is a convention in OMOP that the relationship between a categorical variable concept and the concepts for each of its categories be "has answer". In the OMOP RELATIONSHIP table, there exists a "has answer" RELATIONSHIP record. Together the categorical variable concept, the category concepts and the "has answer" CONCEPT_RELATIONSHIPs that goes between them defines a code list aka "category set" (DDI) in OMOP:

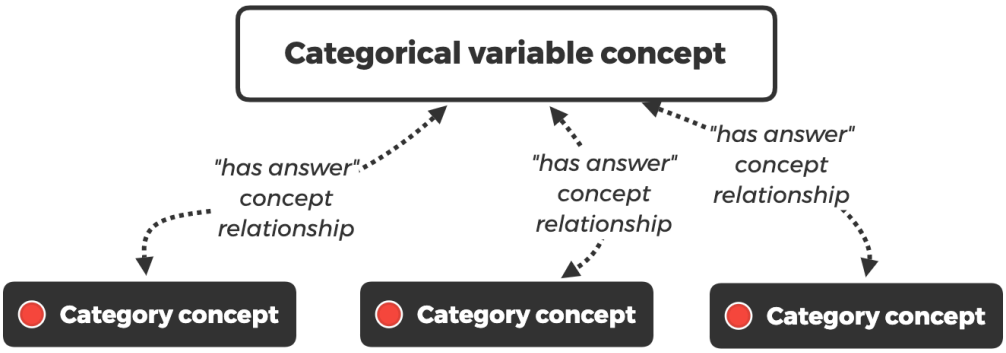


Figure 7: The OMOP codelist aka category set construct

OHDSI USAGI tool has been used to discover these concept_ids both from the OMOP Standardized Vocabularies and a de novo HDSS vocabulary. The INSPIRE Resource Centre includes a tutorial on how to use USAGI to discover the Standardized Vocabularies concept_ids we will need to use in the ETLs.

OHDSI WhiteRabbit

The purpose of WhiteRabbit tool is to scan the source data and then create a metadata document in a spreadsheet format, MS Excel .xlsx file, which includes detailed information on the tables, fields/columns and value types with some basic statistics for each field. This scan report can be used to generate a fake dataset to be used for ETL purposes. The scan report is also imported in the ETL documentation tool, the Rabbit In A Hat for the ETL design and testing purpose.

URL: http://ohdsi.github.io/WhiteRabbit/WhiteRabbit.html#installation_and_support

Steps to scan the ALPHA Residencies table using WhiteRabbit

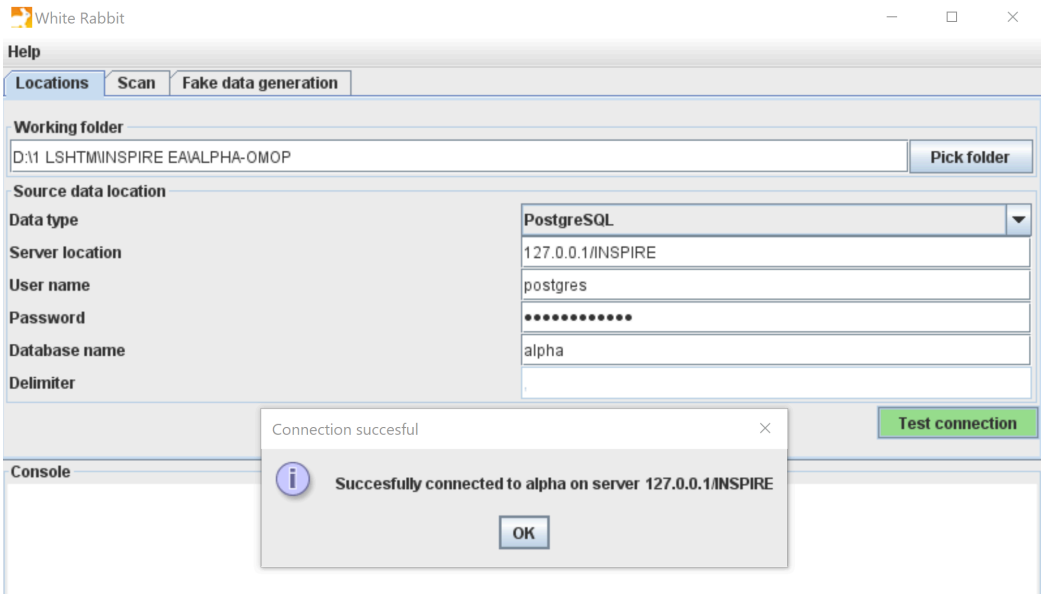
The following steps were performed to generate the scan report and fake data from ALPHA Residencies:

Note here that as part of the data preparation process for this documentation, a dummy ALPHA dataset was migrated to a PostgreSQL database schema.

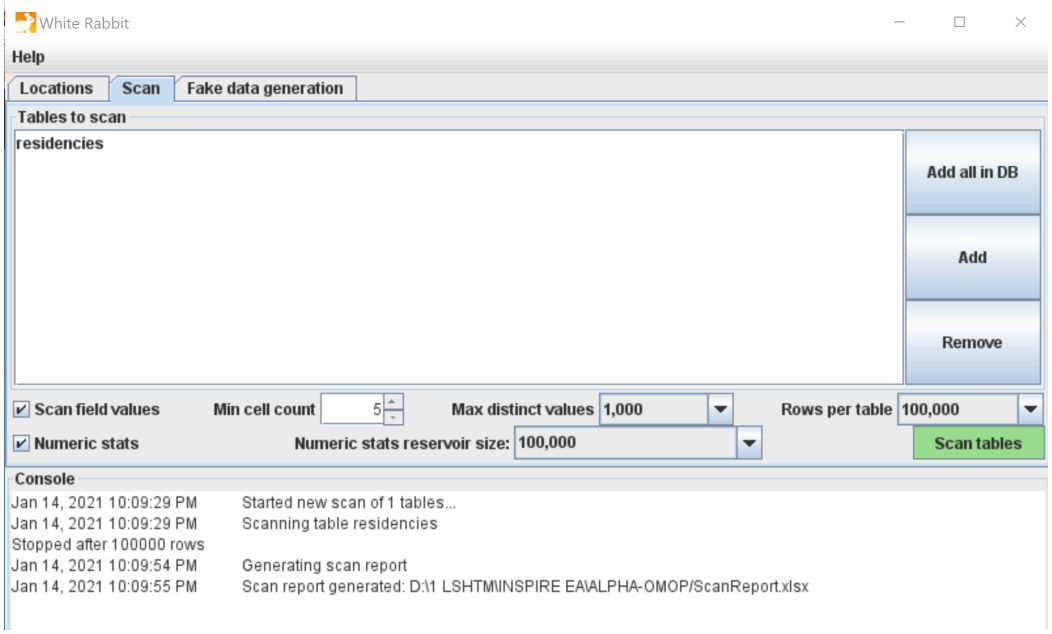
(1) Installed WhiteRabbit version 0.10.2

(https://github.com/OHDSI/WhiteRabbit/releases/tag/v0.10.2)

(2) Launched WhiteRabbit
(3) In the Location tab
    a. Picked the working folder.
    b. Connected to the PostgreSQL database
    c. Tested the connection

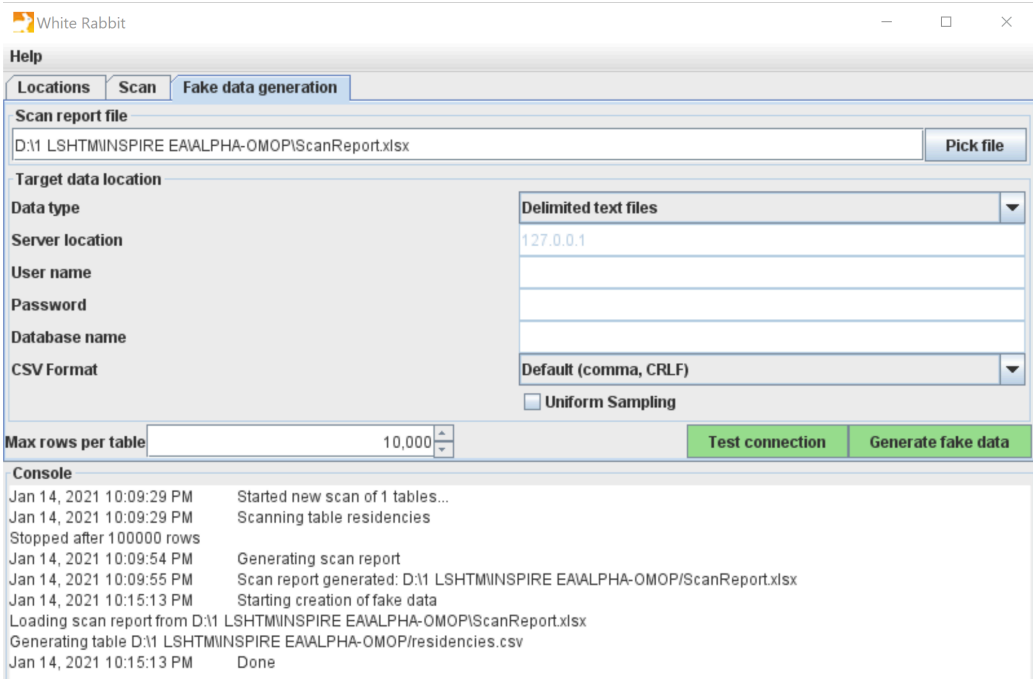


(4) In the Scan tab
   a. Added the Residencies table
   b. Checked the Scan field values and Numeric stats checkbox.
   c. Kept the other fields to default values and then clicked on scan button



(5) In the fake data generation tab
   a. Picked the scan report folder
   b. Saved the fake data in a delimited text file in default csv comma separated file
   c. Tested the connection to ensure that the scan report folder is accessible
   d. The residencies.csv file with fake data of 10,000 rows got created in the working folder

In the following examples, to illustrate the mapping and vocabulary processes in this document, we have not used the fake data generated here, but used a dummy ALPHA dataset from which this fake dataset generation example is demonstrated.



OHDSI Rabbit In A Hat

Rabbit In A Hat uses the scan report produced by White Rabbit, to generate ETL process documentation and not the code needed for implementing ETL process. It is a graphical user interface which allows to connect the source data columns to CMD columns.

URL: http://ohdsi.github.io/WhiteRabbit/RabbitInAHat.html#installation_and_support

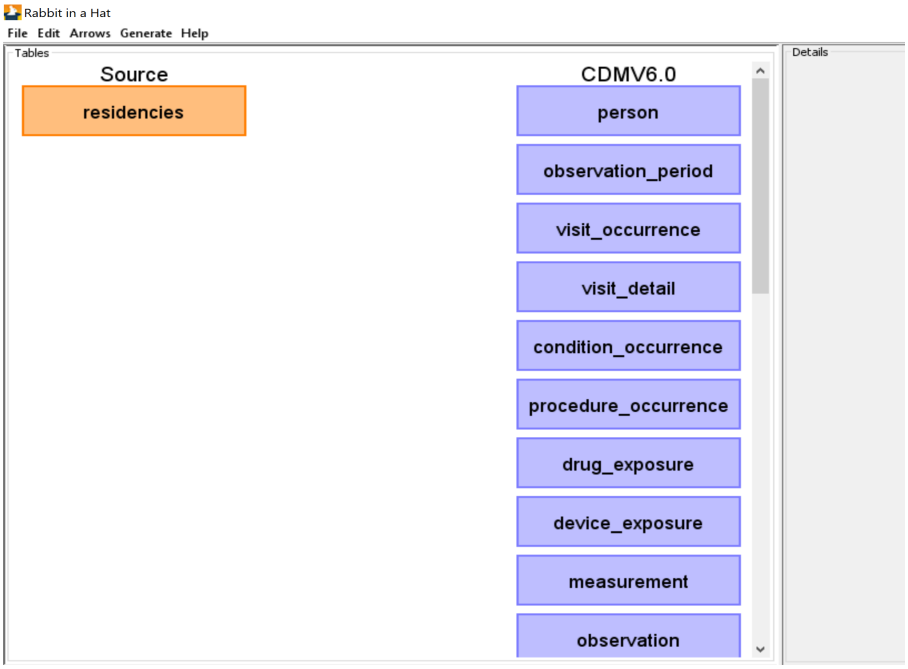
Steps to start the mapping process of ALPHA Residencies to OMOP using Rabbit In A Hat

The following steps were performed to generate the ETL process documentation for data mapping from ALPHA Residencies to OMOP:

(1) Installed Rabbit In A Hat version 0.10.2

(https://github.com/OHDSI/WhiteRabbit/releases/tag/v0.10.2)

(2) Launched Rabbit In A Hat
(3) Opened the Scan Report
    a. Clicked on File from the menu bar
    b. Clicked on Open Scan Report (Shortcut Ctrl+w)
    c. Looked into the scan report folder and opened the scan report



OHDSI Usagi

Usagi is a software tool used to help in the process of mapping codes from a source system into terminologies, preferably standard ones, stored in the Observational Medical Outcomes Partnership (OMOP) Vocabulary. The word Usagi is Japanese for rabbit and was named after the first mapping exercise it was used for; mapping source codes used in a Japanese dataset into OMOP Vocabulary concepts.

URL: https://www.ohdsi.org/web/wiki/doku.php?id=documentation:software:usagi

Steps to map vocabularies from ALPHA to OMOP using Usagi

The following steps were performed to find the vocabulary mapping using Usagi for mapping from ALPHA Residencies to OMOP:

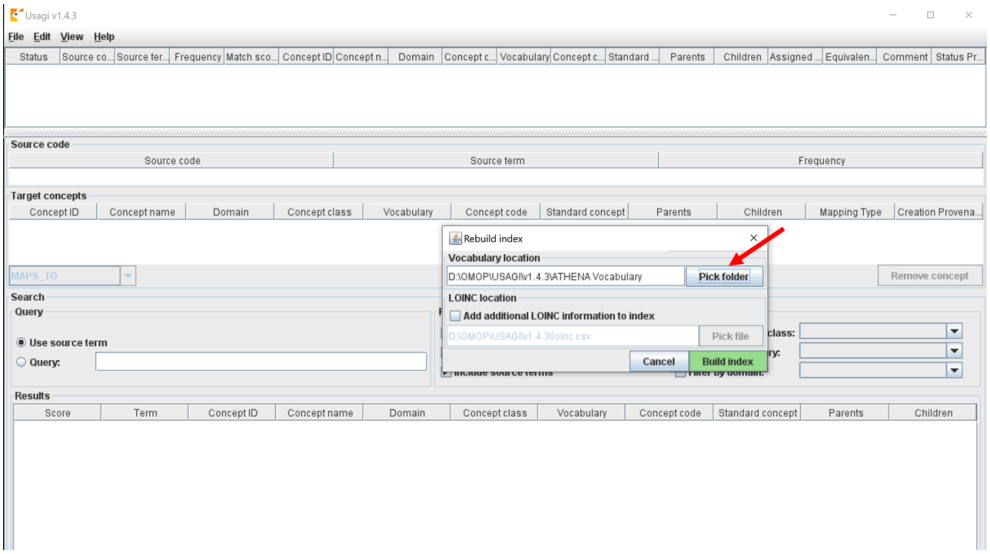
Here in this example, we demonstrate the vocabulary mapping between ALPHA residencies' sex to OMOP's gender variable. Similar type of process can be followed for other vocabulary mappings.

(1) Installed Usagi 1.4.3.

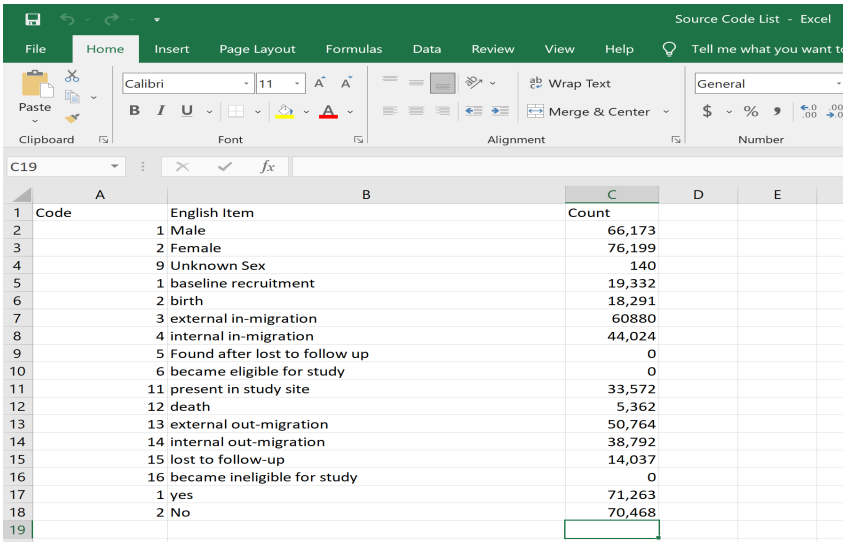
(https://github.com/OHDSI/Usagi/releases)

(2) Launched Usagi
(3) For the first time, it asked for the author's name, you may give any name you wish to and save it

(4) For the first time, built the index using the ATHENA vocabulary I had downloaded from the ATHENA website (https://athena.ohdsi.org/) before step (1).
(5) Picked the folder, where I had stored the ATHENA vocabulary files (.csv)

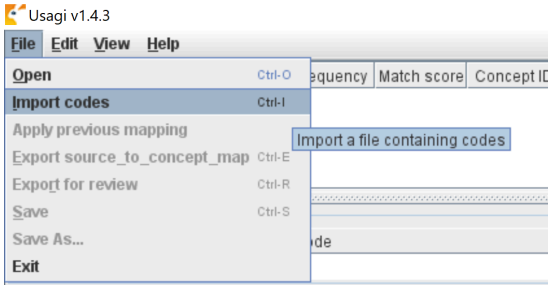


(6) Clicked on Build Index. This step took a while to complete.
(7) If you wish to upgrade the Index which has already been built, then click on View from the Menu bar and then click on Rebuild Index.
(8) After completion of the Index building process, restarted Usagi to use this new index
(9) To map the vocabulary from ALPHA residencies to OMOP, I created the following Excel file as input

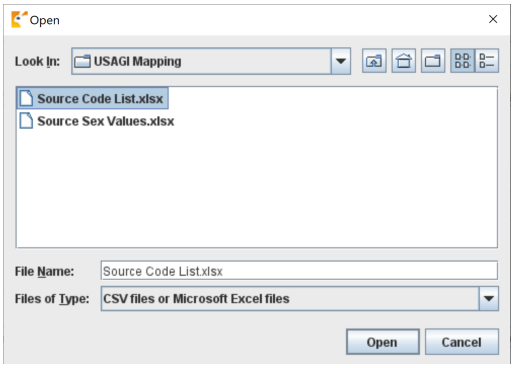


	A	B	C	D	E
1	Code	English Item	Count		
2	1	Male	66,173		
3	2	Female	76,199		
4	9	Unknown Sex	140		
5	1	baseline recruitment	19,332		
6	2	birth	18,291		
7	3	external in-migration	60880		
8	4	internal in-migration	44,024		
9	5	Found after lost to follow up	0		
10	6	became eligible for study	0		
11	11	present in study site	33,572		
12	12	death	5,362		
13	13	external out-migration	50,764		
14	14	internal out-migration	38,792		
15	15	lost to follow-up	14,037		
16	16	became ineligible for study	0		
17	1	yes	71,263		
18	2	No	70,468		
19					

(10) In Usagi, Clicked on File and then Import Codes

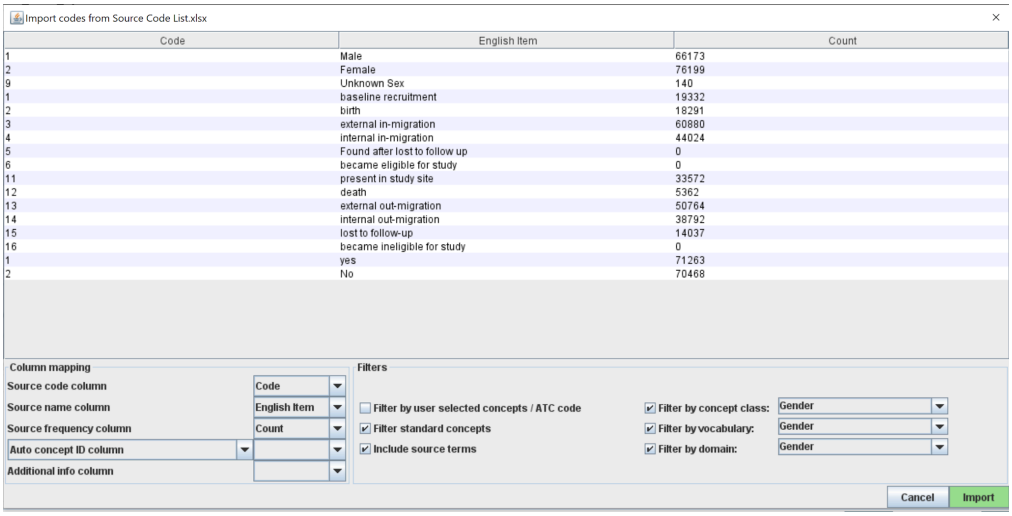


(11) Navigated to the folder where I have stored the ALPHA codes and opened it



(12) Selected the column mapping as follows:

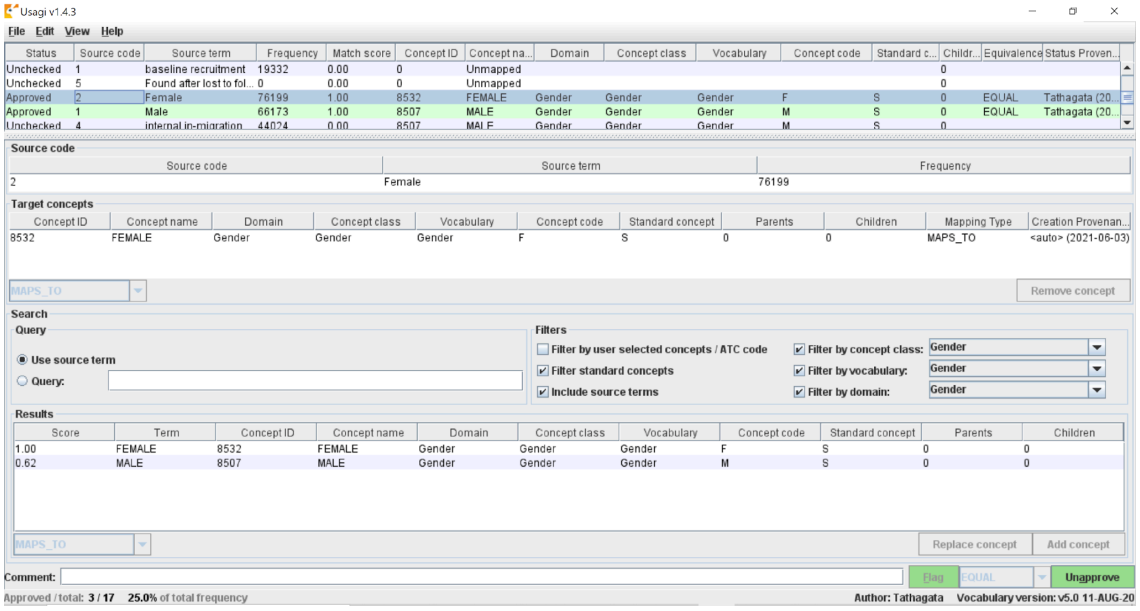
Source code column = Code
Source name Column = English Item
Source frequency column = count
Now, to map the Sex vocabulary, I filtered by Gender and clicked on Import.



(13) Selected the source code 1 where the source term is Male and domain, concept class and vocabulary are Gender and then clicked on Approve

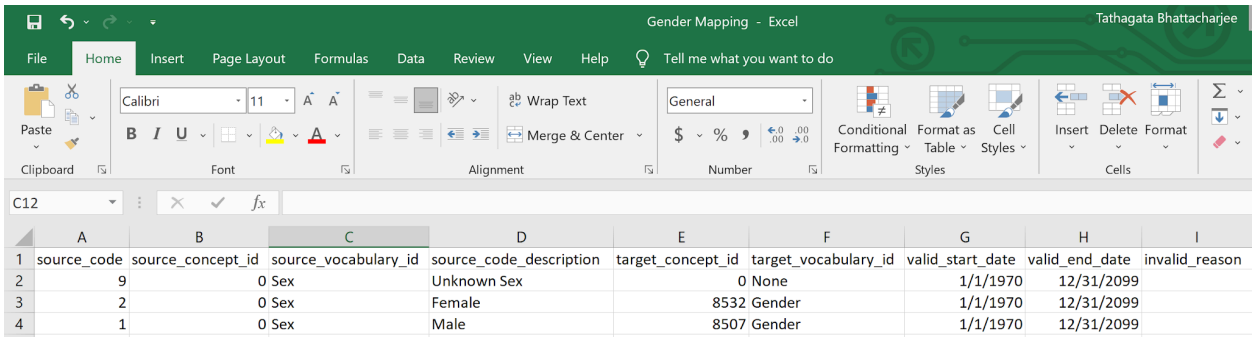
(14) Selected the source code 2 where the source term is Female and domain, concept class and vocabulary are Gender and then clicked on Approve

(15) Selected the source code 9 where the source term is Unknown Sex and domain, concept class and vocabulary are Gender and then clicked on Approve



(16) Exported the Gender mapping.

a. Clicked on File -> Export source_to_concept_map
b. Clicked on Only Approved
c. Typed Sex for the Source vocabulary Id and then clicked on Export
d. Navigated to the folder to store my mapping files and named it Gender Mapping.csv
e. Clicked on Save



	A	B	C	D	E	F	G	H	I
1	source_code	source_concept_id	source_vocabulary_id	source_code_description	target_concept_id	target_vocabulary_id	valid_start_date	valid_end_date	invalid_reason
2	9	0	Sex	Unknown Sex	0	None	1/1/1970	12/31/2099	
3	2	0	Sex	Female	8532	Gender	1/1/1970	12/31/2099	
4	1	0	Sex	Male	8507	Gender	1/1/1970	12/31/2099	

Mapping ALPHA Residencies to OMOP v6.0 CDM

In this section of the document, we will discuss the mapping process from ALPHA residencies (ALPHA Data Spec 6.1) to OMOP v6.0 CDM.

Source Table(s) Structure:

Table: residencies

The following table shows the ALPHA residencies data specification field names with descriptions.

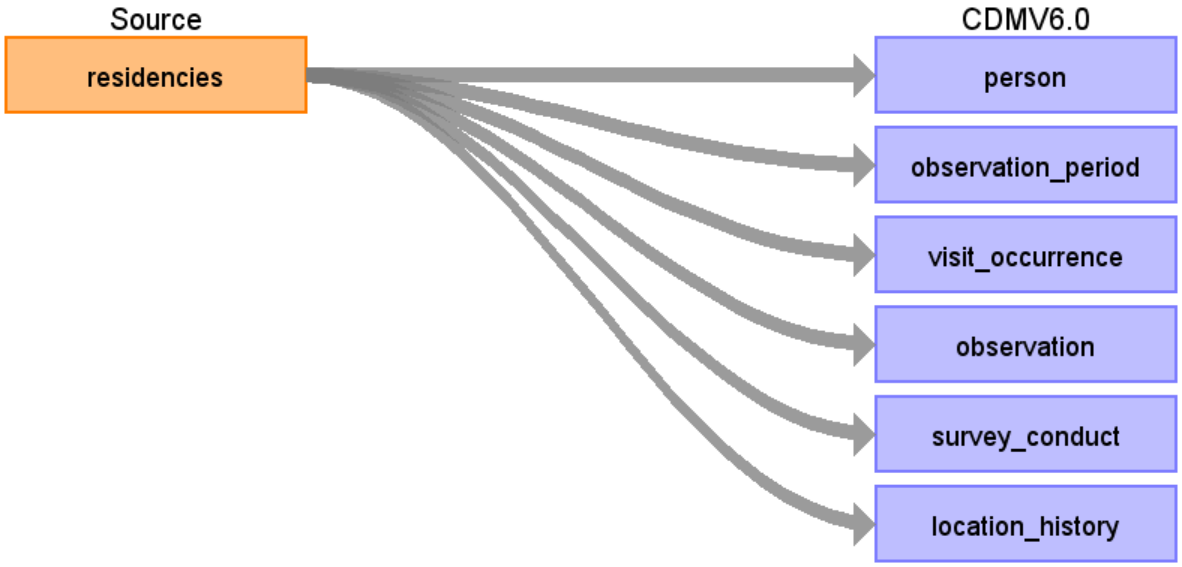
Field	Description	Type	Mapping
study_name	Name of study field site	character varying	
idno	Person ID number - Numeric IDs long integer format, unique for an individual	integer	
hhold_id	Household ID number - Geographical location, should be unique for each household	character varying	
hhold_id_extra	Household ID number - If site has another definition for Households (e.g., social units) include it here	character varying	
sex	Male or female - must not vary between residence episodes 1 = Male | 2 = Female	smallint	FK into CONCEPT table
dob	Date of birth- best estimate - If actual month and day are not known it is OK to impute, e.g., assign to middle of the month or mid-year Must not vary between residence episodes	date	
residence	Type of area within DSS - Aim to distinguish urban / rural, or among rural areas distinguish remote / roadside, or by dominant industry 1 = Rural | 2 = Semi Urban | 3 = Urban	smallint	FK into CONCEPT table
entry_type	A code identifying an episode entry event that has occurred: 1 - baseline recruitment 2 - birth 3 - external in-migration 4 - Internal in-migration 5 - Found after lost to follow up 6 - became eligible for study	smallint	FK into CONCEPT table
entry_date	Date on which the event occurred	date	
entry_type_of_date	Description of how entry_date obtained 1 = Reported by HH informant at interview 2 = Reported by key informant 3 = Imputed	smallint	
entry_obs_date	Observation date - Date on which the event was observed (recorded), also known as surveillance visit date	date	
entry_obs_round	Observation round - Surveillance round when the event was observed (recorded), also known as surveillance round	smallint	
exit_type	A code identifying an episode exit event that has occurred: 11 - present in study site 12 - death 13 - out-migration 14 - Internal out-migration 15 - lost to follow-up 16 - became ineligible for study	smallint	FK into CONCEPT table

exit_date	Date on which the event occurred	date	
exit_type_of_date	Description of how exit_date obtained 1 = Reported by HH informant at interview 2 = Reported by key informant 3 = Imputed	smallint	
exit_obs_date	Observation date - Date on which the event was observed (recorded), also known as surveillance visit date	date	
exit_obs_round	Observation round - Surveillance round when the event was observed (recorded), also known as surveillance round	smallint	

In the course of the mapping a number of fields – the ones that take "categories" in OMOP change their type and become foreign keys into the CONCEPT table. As foreign keys they reference records in the CONCEPT table where each concept gets a unique **concept_id**. In the tables that follow these fields are colored **red**. Then, in two sections that follow these tables both ***best practices*** are discussed in connection with several of these concepts and ***additional concepts*** are introduced in support of the mapping. The additional concepts are part of a de novo vocabulary not found in the standardized vocabulary. These concepts are specific to longitudinal population studies.

An important point to note here is that the nomenclature with respect to visit and observation that is used in ALPHA and OMOP are different. For understanding and used in this mapping process are:
ALPHA observations = OMOP visit and ALPHA events = OMOP observations

Source Data Mapping Approach to CDMV6.0



The ALPHA residencies, which stores the longitudinal population data in the wide format, i.e., the episode format is been mapped to six OMOP CDM tables and show in the diagram above.

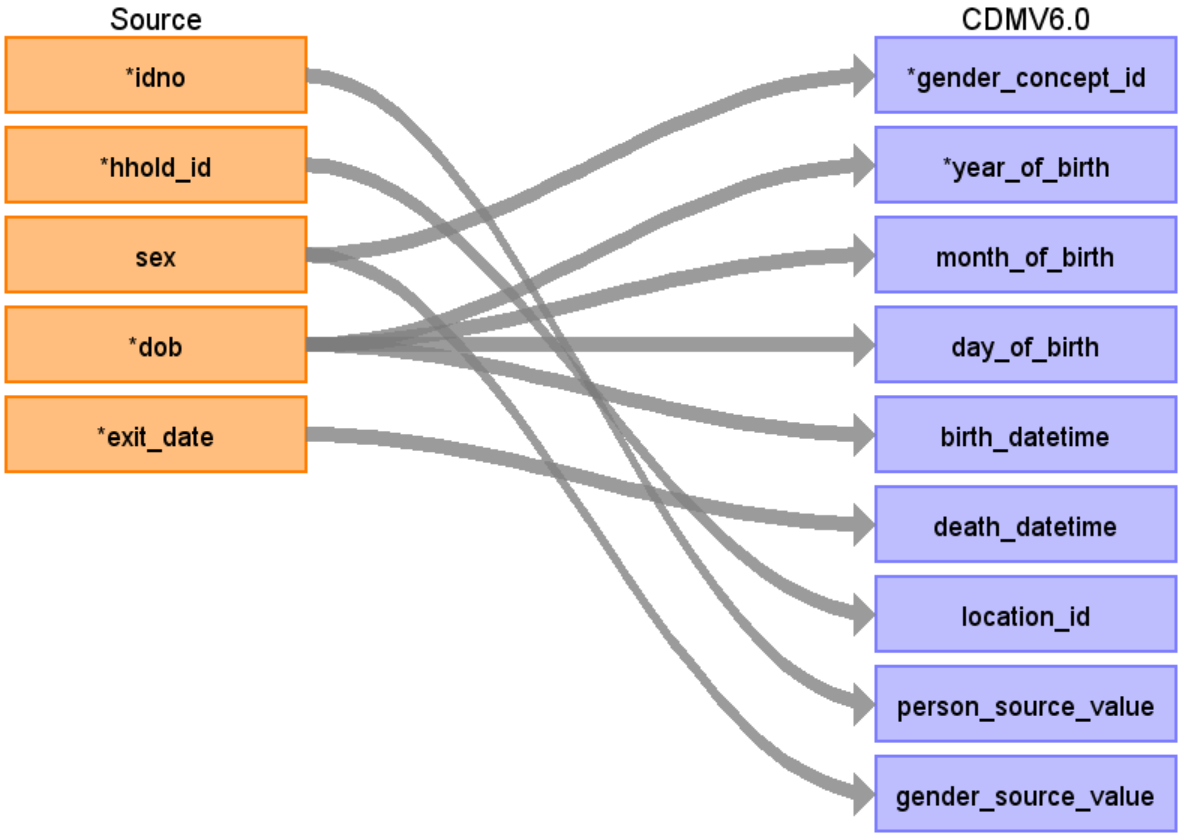
OMOP Table name: person

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

Adaptation for ALPHA to OMOP mapping: This table holds the personal details of individuals mapped from ALPHA residencies table.

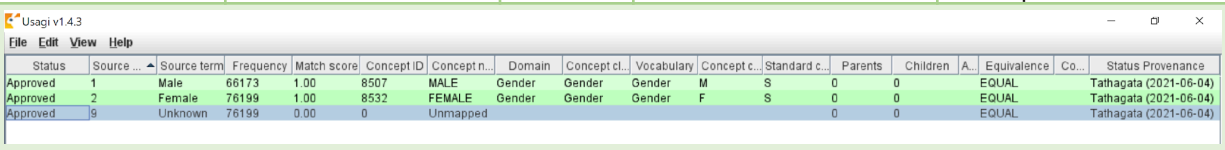
ALPHA Residencies to OMOP Person

Reading from residencies: The person details, i.e., the individual details from the ALPHA residencies are mapped to this table. While migrating data from ALPHA residencies to OMOP person, we have to extract and migrate unique individuals' details as ALPHA residencies store residency episodes and an individual may have multiple episodes. Take the latest entry of episode for this purpose.



Guidelines for Mapping ALPHA Residencies to OMOP Person

Destination Field	Field Description	Source Field	Logic	Comment

person_id	A unique identifier for each person		Generate a sequential number to identify each person uniquely in the database	This is an anonymised id for every person in the database.
gender_concept_id	A foreign key that refers to an identifier in the CONCEPT table for the unique gender of the person	sex	The sex value from ALPHA data specs will be mapped to OMOP standard gender concept id. 1 (Male) = 8507 2 (Female) = 8532 9 (Unknown) = 0 *The unknown gender concept id 8551 is deprecated since 31-Jul-2014*	DETAILS: Domain ID: Gender Concept Class ID: Gender Vocabulary ID: Gender Validity: Valid ConceptStandard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099 Male: Concept ID: 8507 Concept code: M Female: Concept ID: 8532 Concept code: F
				
year_of_birth	The year of birth of the person. For data sources with date of birth, the year is extracted. For data sources where the year of birth is not available, the approximate year of birth is derived based on any age group categorization available	dob	Extract the YEAR part from date of birth (dob)	PostgreSQL query: *SELECT EXTRACT (YEAR FROM dob)* *FROM alpha.residencies;*
month_of_birth	The month of birth of the person. For data sources that provide the precise date of birth, the month is extracted and stored in this field	dob	Extract the MONTH part from date of birth (dob)	PostgreSQL query: *SELECT EXTRACT (MONTH FROM dob)* *FROM alpha.residencies;*
day_of_birth	The day of the month of birth of the person. For data sources that provide the precise date of birth, the day is extracted and stored in this field	dob	Extract the DAY part from date of birth (dob)	PostgreSQL query: *SELECT EXTRACT (DAY FROM dob)* *FROM alpha.residencies;*
birth_datetime	The date and time of birth of the person	dob	Only the DATE part of date of birth (dob) will be mapped here, as ALPHA	Store in *YYYY-MM-DD* format

			does not store the whole TIMESTAMP value	
death_datetime	The date and time of death of the person	exit_date	Extract the event DEATH from exit_type to get the date of death to be mapped to this variable. Only the DATE part will be mapped here, as ALPHA does not store the whole TIMESTAMP value	PostgreSQL query*:* *SELECT exit_date* *FROM alpha.residencies* *WHERE exit_type = 12;* Store in *YYYY-MM-DD* format
race_concept_id	A foreign key that refers to an identifier in the CONCEPT table for the unique race of the person, belonging to the 'Race' vocabulary		Statically imputed. As, all ALPHA sites are in Africa, mostly in rural areas, we set the race by default to African i.e., race_concept_id = 38003600.	DETAILS: Domain ID: Race Concept Class ID: Race Vocabulary ID: Race Concept ID: 38003600 Concept code: 3.03 Validity: Valid Concept: Standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
ethnicity_concept_id	A foreign key that refers to the standard concept identifier in the Standardized Vocabularies for the ethnicity of the person, belonging to the 'Ethnicity' vocabulary		Statically imputed. As all ALPHA sites are in Africa, mostly in rural areas, we can set the ethnicity by default to Not Hispanic or Latino i.e., ethnicity_concept_id = 38003564	DETAILS: Domain ID: Ethnicity Concept Class ID: Ethnicity Vocabulary ID: Ethnicity Concept ID: 38003564 Concept code: Not Hispanic Validity: Valid Concept: Standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
location_id	A foreign key to the place of residency for the person in the location table, where the detailed address information is stored	hhold_id	Store the latest/current hhold_id because for any internal movement(s), the hhold_id change indicates an internal migration, which are tracked in the location table	The location_id here points to the location, i.e., the household identification id, where the person lives within the HDSS.
provider_id	A foreign key to the primary care provider the person is seeing in the provider table			ALPHA residencies do not track this information, so skip it
care_site_id	A foreign key to the site of primary care in the care_site table, where the details of the care site are stored			HDSS is a household survey and care site are not applicable, so skip it
person_source_value	An (encrypted) key derived from the person identifier in the source data. This is necessary when a use case requires a link	idno	The original value of idno as in the ALPHA residencies will be stored here	It is assumed that the idno in ALPHA residencies is already anonymised and thus further encryption is not needed.

	back to the person data at the source dataset			
gender_source_value	The source code for the gender of the person as it appears in the source data. The personâ€™s gender is mapped to a standard gender concept in the Standardized Vocabularies; the original value is stored here for reference	sex	The original value stored in ALPHA residencies for sex variable will be stored here	Male = 1 Female = 2 Unknown Sex = 9
gender_source_concept_id	A foreign key to the gender concept that refers to the code used in the source		No additional gender concept is maintained in ALPHA residencies, so populate it with value 0 as 'No matching concept'	DETAILS Domain ID: Metadata Concept Class: ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
race_source_value	The source code for the race of the person as it appears in the source data. The person race is mapped to a standard race concept in the Standardized Vocabularies and the original value is stored here for reference			ALPHA residencies do not store any race related value, so skip it
race_source_concept_id	A foreign key to the race concept that refers to the code used in the source		ALPHA residencies do not store any race related value, so populate it with value 0 as 'No matching concept'	DETAILS Domain ID: Metadata Concept Class: ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
ethnicity_source_value	The source code for the ethnicity of the person as it appears in the source data. The person ethnicity is			ALPHA residencies do not store any ethnicity related value, so skip it

	mapped to a standard ethnicity concept in the Standardized Vocabularies and the original code is, stored here for reference			
ethnicity_source_concept_id	A foreign key to the ethnicity concept that refers to the code used in the source		ALPHA residencies do not store any ethnicity related value, so populate it with value 0 as 'No matching concept'	DETAILS Domain ID: Metadata Concept Class: ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099

SQL Skeleton codes for ETL: ALPHA Residencies to OMOP Person

Following is a set of SQL skeleton code for reference purpose only for implementation of ETL to migrate data from ALPHA residencies to OMOP person table. This SQL code is generated for PostgreSQL database where data is migrated from the INSPIRE database's alpha schema to cdm schema.

SQL statement to create the OMOP person table in INSPIRE database's cdm schema.

```
CREATE TABLE IF NOT EXISTS cdm.person
(
  person_id integer NOT NULL,
  gender_concept_id character varying(4) NOT NULL,
  year_of_birth double precision NOT NULL,
  month_of_birth double precision,
  day_of_birth double precision,
  birth_datetime timestamp without time zone,
  death_datetime timestamp without time zone,
  race_concept_id double precision NOT NULL,
  ethnicity_concept_id double precision NOT NULL,
  location_id double precision,
  provider_id double precision,
  care_site_id double precision,
  person_source_value integer,
  gender_source_value smallint,
  gender_source_concept_id double precision NOT NULL,
  race_source_value text,
  race_source_concept_id double precision NOT NULL,
  ethnicity_source_value character varying(200),
  ethnicity_source_concept_id double precision NOT NULL,
  CONSTRAINT person_pkey PRIMARY KEY (person_id)
```

```
);
```

SQL statement to create a Sequence to generate the person_id in PostgreSQL

```
CREATE SEQUENCE cdm.person_id_seq;
```

SQL statement to insert data into cdm.person table from alpha.residencies table

```
INSERT INTO cdm.person
(
        person_id,
        gender_concept_id,
        year_of_birth,
        month_of_birth,
        day_of_birth,
        birth_datetime,
        death_datetime,
        race_concept_id,
        ethnicity_concept_id,
        location_id,
        provider_id,
        care_site_id,
        person_source_value,
        gender_source_value,
        gender_source_concept_id,
        race_source_value,
        race_source_concept_id,
        ethnicity_source_value,
        ethnicity_source_concept_id
)
SELECT
DISTINCT ON (idno)
NEXTVAL('cdm.person_id_seq') AS person_id,
CASE WHEN alpha.residencies.sex = 1 THEN 8507
     WHEN alpha.residencies.sex = 2 THEN 8532
     WHEN alpha.residencies.sex = 9 THEN 0
     ELSE 0
 END AS gender_concept_id,
EXTRACT(YEAR FROM alpha.residencies.dob) AS year_of_birth,
EXTRACT(MONTH FROM alpha.residencies.dob) AS month_of_birth,
EXTRACT(DAY FROM alpha.residencies.dob) AS day_of_birth,
residencies.dob AS birth_datetime,
CASE WHEN alpha.residencies.exit_type = 12 THEN alpha.residencies.exit_date
      ELSE NULL
 END AS death_datetime,
38003600 AS race_concept_id,
38003564 AS ethnicity_concept_id,
CAST(alpha.residencies.hhold_id AS INT) AS location_id,
NULL AS provider_id,
NULL AS care_site_id,
```

```
CAST(alpha.residencies.idno AS INT) AS person_source_value,
residencies.sex AS gender_source_value,
o AS gender_source_concept_id,
NULL AS race_source_value,
o AS race_source_concept_id,
NULL AS ethnicity_source_value,
o AS ethnicity_source_concept_id
FROM alpha.residencies
ORDER BY idno, entry_date DESC;
```

OMOP Table name: observation_period

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

Adaptation for ALPHA to OMOP mapping: This table holds the observation period of a person within the HDSS area i.e, the entry and exit from the HDSS area. In cases of outmigration followed by re-entry into the HDSS area, the in-between period when the person was outside the HDSS area would be omitted and thus two episodes would be created. In cases of Death, no further observation period would be created.

ALPHA Residencies to OMOP Observation_Period

Reading from residencies: Each observation of an episode, i.e., the start observation and the end observation for each episode from ALPHA residencies are mapped to this table. Here we are mapping the start and end event dates of an episode as the observation period.



Guidelines for Mapping ALPHA Residencies to OMOP Observation_Period

Destination Field	Field Description	Source Field	Logic	Comment
observation_period_id	A unique identifier for each observation period		Generate a sequential number to identify each observation period uniquely, i.e., assigning a unique number to each episode	
person_id	A foreign key identifier to the person for whom the observation		This will be a foreign key identifier of the person for	Link this with person_id column of OMOP

	period is defined. The demographic details of that person are stored in the person table		whom the observation period is defined	person table using person_source_value
observation_period_start_date	The start date of the observation period for which data are available from the data source	entry_date	Extract the entry_date from ALPHA residency table which will denote the episode start date	Store date in *YYYY-MM-DD* format
observation_period_end_date	The end date of the observation period for which data are available from the data source	exit_date	Extract the exit_date from ALPHA residency table which will denote the episode end date	Store date in *YYYY-MM-DD* format
period_type_concept_id	foreign key identifier to the predefined concept in the Standardized Vocabularies reflecting the source of the observation period information, belonging to the 'Obs Period Type' vocabulary		Statically imputed. ALPHA residencies are created from HDSS (demography - longitudinal population data), so it is mapped to the "Period while enrolled in study" period concept id i.e., period_type_concept_id = 44814723	DETAILS: Domain ID: Type Concept Concept Class ID: Obs Period Type Vocabulary ID: Obs Period Type Concept ID: 44814723 Concept code: OMOP4822292 Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099

SQL Skeleton codes for ETL: ALPHA Residencies to OMOP Observation_Period

Following is a set of SQL skeleton code for reference purpose only for implementation of ETL to migrate data from ALPHA residencies to OMOP observation_period table. This SQL code is generated for PostgreSQL database where data is migrated from the INSPIRE database's alpha schema to cdm schema.

SQL statement to create the OMOP observation_period table in INSPIRE database's cdm schema.

```
CREATE TABLE IF NOT EXISTS cdm.observation_period
(
  observation_period_id bigint NOT NULL,
  person_id integer NOT NULL,
  observation_period_start_date date NOT NULL,
  observation_period_end_date date NOT NULL,
  period_type_concept_id integer NOT NULL,
  CONSTRAINT xpk_observation_period PRIMARY KEY (observation_period_id)
);
```

SQL statement to create a Sequence to generate the observation_period_id in PostgreSQL

```
CREATE SEQUENCE cdm.observation_period_id_seq;
```

SQL statement to insert data into cdm.observation_period_id table from alpha.residencies table

```
INSERT INTO cdm.observation_period
(
  observation_period_id,
        person_id,
  observation_period_start_date,
  observation_period_end_date,
  period_type_concept_id
)
SELECT
  NEXTVAL('cdm.observation_period_id_seq')  AS observation_period_id,
  cdm.person.person_id AS person_id,
  alpha.residencies.entry_date AS observation_period_start_date,
  alpha.residencies.exit_date AS observation_period_end_date,
  44814723 AS period_type_concept_id
FROM alpha.residencies
INNER JOIN cdm.person
ON CAST(cdm.person.person_source_value AS INT) = CAST(alpha.residencies.idno AS INT);
```

OMOP Table name: visit_occurrence

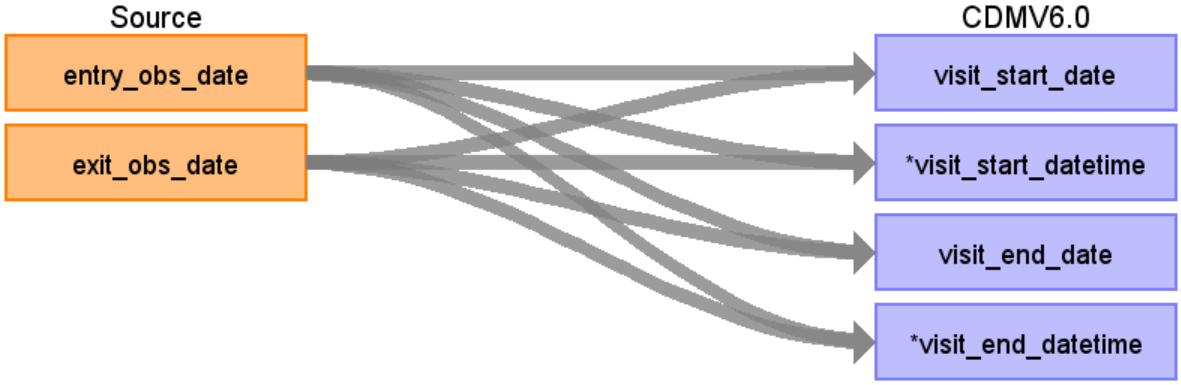
OMOP definition: The VISIT_OCCURRENCE table contains the spans of time a Person continuously receives medical services from one or more providers at a Care Site in a given setting within the health care system. Visits are classified into 4 settings: outpatient care, inpatient confinement, emergency room,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

Adaptation for ALPHA to OMOP mapping: This table holds the visits that were made to households as part of the HDSS survey. The ALPHA residencies table holds the observation date i.e., the interview date, which we would map here as the visit date and thus the visit start and end date would be the same for a particular visit and this is in line with the definition of visit_end_date on OMOP table. So, for every episode, there will be two sets of visit start and end dates. No TIMESTAMP values are available in ALPHA residencies data spec so only DATE value will be populated.

Here the ALPHA residencies wide format (episode format) will get converted into long format (event format).

ALPHA Residencies to OMOP Visit_Occurence

Reading from residencies:



Guidelines for Mapping ALPHA Residencies to OMOP Visit_Occurrence

Destination Field	Field Description	Source Field	Logic	Comment
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.		Generate a sequential number to identify each visit occurrence uniquely, i.e., assigning a unique number to each household visit	
person_id			This will be a foreign key identifier of the person for whom the visit occurrence is defined	Link this with person_id column of OMOP person table using person_source_value
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.		Statically imputed. The visit_concept_id to be used is “Home Visit” visit_concept_id = 581476	DETAILS: Domain ID: Visit Concept Class ID: Visit Vocabulary ID: Visit Concept ID: 581476 Concept code: OMOP4822459 Validity: Valid Concept: Standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
visit_start_date	For inpatient visits, the start date is typically the admission date. For outpatient visits the	entry_obs_date exit_obs_date	Map the entry_obs_date or exit_obs_date to populate this variable	Convert the wide format (episode format) to long format (event format) Store date in *YYYY-MM-DD* format

	start date and end date will be the same.			
visit_start_datetime	If no time is given for the start date of a visit, set it to midnight (00:00:0000).		Map the entry_obs_date or exit_obs_date in TIMESTAMP format. If the source field value is NULL then take a default date as '9999-12-31 00:00:00'.	This is a mandatory field in OMOP CDM. Convert the wide format (episode format) to long format (event format). Store date in *YYYY-MM-DD 00:00:00* format
visit_end_date	For inpatient visits the end date is typically the discharge date.	entry_obs_date exit_obs_date	Map the entry_obs_date or exit_obs_date to populate this variable	Convert the wide format (episode format) to long format (event format) Store date in *YYYY-MM-DD* format
visit_end_datetime	If no time is given for the end date of a visit, set it to midnight (00:00:0000).		Map the entry_obs_date or exit_obs_date in TIMESTAMP format. If the source field value is NULL then take a default date as '9999-12-31 00:00:00'.	This is a mandatory field in OMOP CDM. Convert the wide format (episode format) to long format (event format). Store date in *YYYY-MM-DD 00:00:00* format
visit_type_concept_id	Use this field to understand the provenance of the visit record, or where the record comes from.		Statically imputed. The visit_type_concept_id to type concept "Case Report Form" visit_type_concept_id = 32809	DETAILS: Domain ID: Type Concept Concept Class ID: Type Concept Vocabulary ID: Type Concept Concept ID: 32809 Concept code: OMOP4976882 Validity: Valid Concept: Standard Valid start: 20-Aug-2020 Valid end: 31-Dec-2099
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 source, this can be reflected in the event tables			ALPHA residencies do not track this information, so skip it
care_site_id	This field provides information about the care site where the visit took place.			HDSS is a household survey and care site are not applicable, so skip it

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.)			No source value defines a visit, it is identified as an Interview. ALPHA does not store the interview id, so skip it
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. If not available set to 0.		Not applicable, so populate it with value 0 as 'No matching concept'	DETAILS Domain ID: Metadata Concept Class: ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
admitting_source_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.		Not applicable, so populate it with value 0 as 'No matching concept'	DETAILS Domain ID: Metadata Concept Class: ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
admitting_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.			Not applicable
discharge_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 discharged to home or sent to a long-term care facility, for example.		Not applicable, so populate it with value 0 as 'No matching concept'	DETAILS Domain ID: Metadata Concept Class: ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard

				Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
discharge_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.			Not applicable
preceding_visit_occurrence_id	Use this field to find the visit that occurred for the person prior to the given visit. There could be a few days or a few years in between.			Populate it with the immediate earlier (preceding) visit_occurence_id

SQL Skeleton codes for ETL: ALPHA Residencies to OMOP Visit_Occerence

Following is a set of SQL skeleton code for reference purpose only for implementation of ETL to migrate data from ALPHA residencies to OMOP visit_occurence table. This SQL code is generated for PostgreSQL database where data is migrated from the INSPIRE database's alpha schema to cdm schema.

SQL statement to create the OMOP visit_occurence table in INSPIRE database's cdm schema

```
CREATE TABLE IF NOT EXISTS cdm.visit_occurrence
(
    visit_occurrence_id bigint NOT NULL,
    person_id bigint NOT NULL,
    visit_concept_id integer NOT NULL,
    visit_start_date date,
    visit_start_datetime timestamp without time zone NOT NULL,
    visit_end_date date,
    visit_end_datetime timestamp without time zone NOT NULL,
    visit_type_concept_id integer NOT NULL,
    provider_id bigint,
    care_site_id bigint,
    visit_source_value character varying(50),
    visit_source_concept_id integer NOT NULL,
    admitting_source_concept_id integer NOT NULL,
    admitting_source_value character varying(50),
    discharge_to_source_value character varying(50),
    discharge_to_concept_id integer NOT NULL,
    preceding_visit_occurrence_id bigint,
```

```
  CONSTRAINT xpk_visit_occurrence PRIMARY KEY (visit_occurrence_id)
);
```

SQL statement to create a Sequence to generate the visit_occurence_id in PostgreSQL

```
CREATE SEQUENCE cdm.visit_occurrence_id_seq;
```

SQL statement to insert data into cdm.person table from alpha.residencies table

```
--Insert the episode start visit dates to convert the wide format to long format
INSERT INTO cdm.visit_occurrence
(
  visit_occurrence_id,
  person_id,
  visit_concept_id,
 visit_start_date,
  visit_start_datetime,
  visit_end_date,
  visit_end_datetime,
  visit_type_concept_id,
  provider_id,
  care_site_id,
  visit_source_value,
  visit_source_concept_id,
  admitting_source_concept_id,
  admitting_source_value,
  discharge_to_concept_id,
  discharge_to_source_value,
  preceding_visit_occurrence_id
)
SELECT
NEXTVAL('cdm.visit_occurrence_id_seq')  AS visit_occurrence_id,
cdm.person.person_id AS person_id,
581476 AS visit_concept_id,
residencies.entry_obs_date AS visit_start_date,
CASE WHEN residencies.entry_obs_date IS NULL THEN '9999-12-31 00:00:00'
      ELSE residencies.entry_obs_date
END AS visit_start_datetime,
residencies.entry_obs_date AS visit_end_date,
CASE WHEN residencies.entry_obs_date IS NULL THEN '9999-12-31 00:00:00'
      ELSE residencies.entry_obs_date
END AS visit_end_datetime,
32809 AS visit_type_concept_id,
NULL AS provider_id,
NULL AS care_site_id,
NULL AS visit_source_value,
0 AS visit_source_concept_id,
0 AS admitting_source_concept_id,
NULL AS admitting_source_value,
```

```
o AS discharge_to_concept_id,
NULL AS discharge_to_source_value,
NULL AS preceding_visit_occurrence_id
FROM alpha.residencies
INNER JOIN cdm.person
ON CAST(cdm.person.person_source_value AS INT) = CAST(alpha.residencies.idno AS INT);

--Insert the episode end visit dates to convert the wide format to long format
INSERT INTO cdm.visit_occurrence
(
   visit_occurrence_id,
  person_id,
   visit_concept_id,
  visit_start_date,
   visit_start_datetime,
   visit_end_date,
   visit_end_datetime,
   visit_type_concept_id,
   provider_id,
   care_site_id,
   visit_source_value,
   visit_source_concept_id,
   admitting_source_concept_id,
   admitting_source_value,
   discharge_to_concept_id,
   discharge_to_source_value,
   preceding_visit_occurrence_id
)
SELECT
NEXTVAL('cdm.visit_occurrence_id_seq')  AS visit_occurrence_id,
cdm.person.person_id AS person_id,
581476 AS visit_concept_id,
residencies.exit_obs_date AS visit_start_date,
CASE WHEN residencies.exit_obs_date IS NULL THEN '9999-12-31 00:00:00'
      ELSE residencies.exit_obs_date
END AS visit_start_datetime,
residencies.exit_obs_date AS visit_end_date,
CASE WHEN residencies.exit_obs_date IS NULL THEN '9999-12-31 00:00:00'
      ELSE residencies.exit_obs_date
END AS visit_start_datetime,
32809 AS visit_type_concept_id,
NULL AS provider_id,
NULL AS care_site_id,
NULL AS visit_source_value,
o AS visit_source_concept_id,
o AS admitting_source_concept_id,
NULL AS admitting_source_value,
o AS discharge_to_concept_id,
NULL AS discharge_to_source_value,
NULL AS preceding_visit_occurrence_id
```

```
FROM alpha.residencies
INNER JOIN cdm.person
ON CAST(cdm.person.person_source_value AS INT) = CAST(alpha.residencies.idno AS INT);
```

OMOP Table name: observation

OMOP definition: 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.

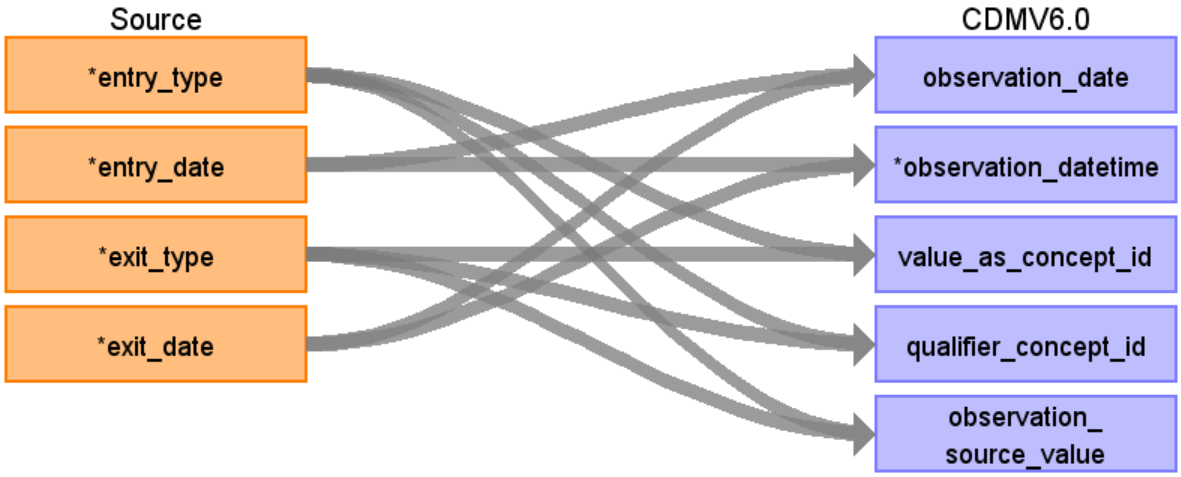
Adaptation for ALPHA to OMOP mapping: Here Observation means the event that was recorded in each visit.

This might seem a bit misnomer with respect to the terminologies used in HDSS where observation means the interview but here, we have interpreted interview (observation in HDSS nomenclature) as VISIT and have put the event that was recorded during the VISIT as OBSERVATION.

Here the ALPHA residencies wide format (episode format) will get converted into long format (event format).

ALPHA Residencies to OMOP Observation

Reading from residencies:



Guidelines for Mapping ALPHA Residencies to OMOP Observation

Destination Field	Field Description	Source Field	Logic	Comment

observation_id	A unique identifier for each observation		Generate a sequential number to identify HDSS event that are recorded at each visit	
person_id	A foreign key identifier to the Person about whom the observation was recorded. The demographic details of that Person are stored in the PERSON table		This will be a foreign key identifier of the person for whom the observation is defined.	Link this with person_id column of OMOP person table using person_source_value
observation_concept_id	A foreign key to the standard observation concept identifier in the Standardized Vocabularies		Statically imputed. The HDSS observes the population of a defined area and thus the observation concept here is taken as 'Population'. observation_concept_id = 4295659	DETAILS: Domain ID: Observation Concept Class ID: Social Context Vocabulary ID: SNOMED Concept ID: 4295659 Concept code: 385436007 Validity: Valid Concept: Standard Synonyms: Population (social concept) Valid start: 31-Jan-2003 Valid end: 31-Dec-2099
observation_date	The date of the observation	entry_date exit_date	Map the entry_date or exit_date to populate this variable	Convert the wide format (episode format) to long format (event format) Store date in *YYYY-MM-DD* format
observation_datetime	The date and time of the observation	entry_date exit_date	Map the entry_obs_date or exit_obs_date in TIMESTAMP format. If the source field value is NULL then take a default date as '9999-12-31 00:00:00'.	This is a mandatory field in OMOP CDM. Convert the wide format (episode format) to long format (event format). Store date in *YYYY-MM-DD 00:00:00* format
observation_type_concept_id	A foreign key to the predefined concept identifier in the Standardized Vocabularies reflecting the type of the observation		Statically imputed. The observation_type_concept_id to be used is "Observation recorded from a survey" observation_type_concept_id = 45905771	DETAILS: Domain ID: Type Concept Concept Class ID: Observation Type Vocabulary ID: Observation Type Concept ID: 45905771 Concept code: OMOP4822321 Validity: Valid

				Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
value_as_number	The observation result stored as a number. This is applicable to observations where the result is expressed as a numeric value		Values are not stored as number, but as a concept	Not applicable
value_as_string	The observation result stored as a string. This is applicable to observations where the result is expressed as verbatim text		Values are not stored as string, but as a concept	Not applicable
value_as_concept_id	A foreign key to an observation result stored as a Concept ID. This is applicable to observations where the result can be expressed as a Standard Concept from the Standardized Vocabularies (e.g., positive/negative, present/absent, low/high, etc.)	entry_type exit_type	Baseline recruitment (Source code = 1): value_as_concept_id = 46235144 Birth (Source code = 2): value_as_concept_id = 4216316 Death (Source code = 12): value_as_concept_id = 4306655 External inmigration & Internal inmigration (Source code = 3,4): value_as_concept_id = 4089508 External outmigration & Internal outmigration (Source code = 13,14): value_as_concept_id = 4114667 value_as_concept_id to be created at a later stage: Found after lost to follow up (Source code = 5) Became eligible for study (Source code = 6) Present in study site (Source code = 11)	These vocabulary mapping is not the final set to be used here. The process is underway to develop a new set of vocabularies for demographic events in longitudinal studies.

			Lost to follow-up (Source code = 15) Became ineligible for study (Source code = 16) As of now, since it is a mandatory field, the concept_id is populated with value 0 as 'No matching concept' for the remaining source values	
qualifier_concept_id	A foreign key to a Standard Concept ID for a qualifier (e.g., severity of drug-drug interaction alert)	entry_type exit_type	If the value_as_concept_id is 'In' (value_concept_id = 4089508) or 'Out' (value_concept_id = 4114667), then we qualify it as Migration. qualifier_concept_id = 44804024 Note: here the qualifier is used to define the In and Out value as Migration type	DETAILS: Domain ID: Observation Concept Class ID: Qualifier Value Vocabulary ID: SNOMED Concept ID: 44804024 Concept code: 723981000000108 Validity: Valid Concept: Standard Synonyms: Migration (qualifier value) Valid start: 01-Oct-2010 Valid end: 31-Dec-2099
unit_concept_id	A foreign key to a Standard Concept ID of measurement units in the Standardized Vocabularies		Statically imputed. The unit_concept_id to be used is "Individual" unit_concept_id = 4299438	DETAILS: Domain ID: Observation Concept Class ID: Social Context Vocabulary ID: SNOMED Concept ID4299438 Concept code: 385435006 Validity: Valid Concept: Standard Synonyms: Individual (person) Valid start: 31-Jan-2003 Valid end: 31-Dec-2099
provider_id	A foreign key to the provider in the PROVIDER table who was responsible for making the observation			Not applicable
visit_occurrence_id	A foreign key to the visit in the VISIT_OCCURRE		Link it as foreign key to visit_occurence_id form the	The visit_occurrence_id in the visit_occurrence

	NCE table during which the observation was recorded		visit_occurence OMOP table	table links both the residencies and the HIV testing observation details, but with a separate set of ids distinguished by the visit_type_concept_id = 32809
visit_detail_id	A foreign key to the visit in the VISIT_DETAIL table during which the observation was recorded			Not applicable
observation_source_value	The observation code as it appears in the source data. This code is mapped to a Standard Concept in the Standardized Vocabularies and the original code is, stored here for reference	exit_type entry_type	See SQL code below for the logic. 1 - baseline recruitment 2 - birth 3 - external in-migration 4 - Internal in-migration 5 - Found after lost to follow up 6 - became eligible for study 11 - present in study site 12 - death 13 - out-migration 14 - Internal out-migration 15 - lost to follow-up 16 - became ineligible for study	Map the entry_type or exit_type to populate this variable. Use the labels and not the numeric code. Here the ALPHA residencies wide format (episode format) will get converted into long format (event format)
observation_source_concept_id	A foreign key to a Concept that refers to the code used in the source		Not applicable, so populate it with value 0 as 'No matching concept'	DETAILS Domain ID: Metadata Concept Class: ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
unit_source_value	The source code for the unit as it appears in the source data. This code is mapped to a standard unit concept in the Standardized Vocabularies and		Statically imputed. The unit of data collection is "Individual"	

	the original code is, stored here for reference			
qualifier_source_value	The source value associated with a qualifier to characterize the observation			Not applicable
observation_event_id	A foreign key to an event table (e.g., PROCEDURE_OCCURRENCE_ID)			Not applicable
obs_event_field_concept_id	A foreign key that refers to a Standard Concept identifier in the Standardized Vocabularies referring to the field represented in the OBSERVATION_EVENT_ID		observation_event_id & obs_event_field_concept_id together form a foreign key into the SURVEY_CONDUCT table.	They associate a question/response OBSERVATION with a survey instance administered to a PERSON.
value_as_datetime	The observation result stored as a datetime value. This is applicable to observations where the result is expressed as a point in time			Not applicable

SQL Skeleton codes for ETL: ALPHA Residencies to OMOP Observation

Following is a set of SQL skeleton code for reference purpose only for implementation of ETL to migrate data from ALPHA residencies to OMOP observation table. This SQL code is generated for PostgreSQL database where data is migrated from the INSPIRE database's alpha schema to cdm schema.

SQL statement to create the OMOP observation table in INSPIRE database's cdm schema

```
CREATE TABLE IF NOT EXISTS cdm.observation
(
  observation_id bigint NOT NULL,
  person_id bigint NOT NULL,
  observation_concept_id integer NOT NULL,
  observation_date date,
  observation_datetime timestamp without time zone NOT NULL,
  observation_type_concept_id integer NOT NULL,
  value_as_number numeric,
  value_as_string character varying(60),
  value_as_concept_id integer,
```

```
  qualifier_concept_id integer,
  unit_concept_id integer,
  provider_id bigint,
  visit_occurrence_id bigint,
  visit_detail_id bigint,
  observation_source_value character varying(50),
  observation_source_concept_id integer NOT NULL,
  unit_source_value character varying(50),
  qualifier_source_value character varying(50),
  observation_event_id bigint,
  obs_event_field_concept_id integer NOT NULL,
  value_as_datetime timestamp without time zone,
  CONSTRAINT xpk_observation PRIMARY KEY (observation_id)
);
```

SQL statement to create a Sequence to generate the observation_id in PostgreSQL

```
CREATE SEQUENCE cdm.observation_id_seq;
```

SQL statement to insert data into cdm.observation table from alpha.residencies table

```
--Insert the episode start observation (Episode entry observation) to convert the wide format to long format
INSERT INTO cdm.observation
(
  observation_id,
  person_id,
  observation_concept_id,
  observation_date,
  observation_datetime,
  observation_type_concept_id,
  value_as_number,
  value_as_string,
  value_as_concept_id,
  qualifier_concept_id,
  unit_concept_id,
  provider_id,
  visit_occurrence_id,
  visit_detail_id,
  observation_source_value,
  observation_source_concept_id,
  unit_source_value,
  qualifier_source_value,
  observation_event_id,
  obs_event_field_concept_id,
  value_as_datetime
)
SELECT
  NEXTVAL('cdm.observation_id_seq')  AS observation_id,
```

```
    cdm.person.person_id AS person_id,
            4295659 AS observation_concept_id,
    residencies.entry_date AS observation_date,
   CASE WHEN alpha.residencies.entry_date IS NULL THEN '9999-12-31 00:00:00'
         ELSE   alpha.residencies.entry_date

  END AS observation_datetime,
    45905771 AS observation_type_concept_id,
    NULL AS value_as_number,
    NULL AS value_as_string,
    CASE WHEN alpha.residencies.entry_type = 1 THEN 46235144
            WHEN alpha.residencies.entry_type = 2 THEN 4216316
            WHEN alpha.residencies.entry_type = 3 THEN 4089508
            WHEN alpha.residencies.entry_type = 4 THEN 4089508
             ELSE 0
  END AS value_as_concept_id,
    CASE WHEN alpha.residencies.entry_type = 3 THEN 44804024
               WHEN alpha.residencies.entry_type = 4 THEN 44804024
               ELSE NULL
            END AS qualifier_concept_id,
    4299438 AS unit_concept_id,
    NULL AS provider_id,
      NULL   AS visit_occurrence_id, -- Link it as foreign key to visit_occurence_id form the visit_occurence
OMOP table
    NULL AS visit_detail_id,
    CASE WHEN alpha.residencies.entry_type = 1 THEN '1 - baseline recruitment'
            WHEN alpha.residencies.entry_type = 2 THEN '2 - birth'
            WHEN alpha.residencies.entry_type = 3 THEN '3 - external in-migration'
            WHEN alpha.residencies.entry_type = 4 THEN '4 - Internal in-migration'
            WHEN alpha.residencies.entry_type = 5 THEN '5 - Found after lost to follow up '
            WHEN alpha.residencies.entry_type = 6 THEN '6 - became eligible for study'
     END AS observation_source_value,
    0 AS observation_source_concept_id,
    'Individual' AS unit_source_value,
    NULL AS qualifier_source_value,
    NULL AS observation_event_id,
    0 AS obs_event_field_concept_id,
    NULL AS value_as_datetime
  FROM alpha.residencies
  INNER JOIN cdm.person
  ON CAST(cdm.person.person_source_value AS INT) = CAST(alpha.residencies.idno AS INT);

  --Insert the episode end observation (Episode exit observation) to convert the wide format to long format
  INSERT INTO cdm.observation
  (
    observation_id,
    person_id,
    observation_concept_id,
    observation_date,
    observation_datetime,
```

```
    observation_type_concept_id,
    value_as_number,
    value_as_string,
    value_as_concept_id,
    qualifier_concept_id,
    unit_concept_id,
    provider_id,
    visit_occurrence_id,
    visit_detail_id,
    observation_source_value,
    observation_source_concept_id,
    unit_source_value,
    qualifier_source_value,
    observation_event_id,
    obs_event_field_concept_id,
    value_as_datetime
)
SELECT
    NEXTVAL('cdm.observation_id_seq')  AS observation_id,
    cdm.person.person_id AS person_id,
            4295659 AS observation_concept_id,
    residencies.exit_date AS observation_date,
    CASE WHEN alpha.residencies.exit_date IS NULL THEN '9999-12-31 00:00:00'
          ELSE  alpha.residencies.exit_date
END AS observation_datetime,
    45905771 AS observation_type_concept_id,
    NULL AS value_as_number,
    NULL AS value_as_string,
   CASE WHEN alpha.residencies.exit_type = 12 THEN 4306655
         WHEN alpha.residencies.exit_type = 13 THEN 4114667
         WHEN alpha.residencies.exit_type = 14 THEN 4114667
         ELSE 0
END AS value_as_concept_id,
CASE WHEN alpha.residencies.exit_type = 13 THEN 44804024
        WHEN alpha.residencies.exit_type = 14 THEN 44804024
        ELSE NULL
END AS qualifier_concept_id,
    4299438 AS unit_concept_id,
    NULL AS provider_id,
      NULL  AS visit_occurrence_id, -- Link it as foreign key to visit_occurence_id form the visit_occurence
OMOP table
    NULL AS visit_detail_id,
    CASE WHEN alpha.residencies.exit_type = 11 THEN '11 - present in study site'
                WHEN alpha.residencies.exit_type = 12 THEN '12 - death'
                WHEN alpha.residencies.exit_type = 13 THEN '13 - out-migration'
                 WHEN alpha.residencies.exit_type = 14 THEN '14 - Internal out-migration'
                 WHEN alpha.residencies.exit_type = 15 THEN '15 - lost to follow-up'
                 WHEN alpha.residencies.exit_type = 16 THEN '16 - became ineligible for study'
END AS observation_source_value,
    0 AS observation_source_concept_id,
```

```
  'Individual' AS unit_source_value,
  NULL AS qualifier_source_value,
  NULL AS observation_event_id,
  o AS obs_event_field_concept_id,
  NULL AS value_as_datetime
FROM alpha.residencies
INNER JOIN cdm.person
ON CAST(cdm.person.person_source_value AS INT) = CAST(alpha.residencies.idno AS INT);
```

SQL statement to update the visit_occurrence_id by linking it to the visit_occurrence table

```
UPDATE cdm.observation
SET visit_occurrence_id = visit_occurrence.visit_occurrence_id
FROM cdm.visit_occurrence
WHERE cdm.observation.person_id = cdm.visit_occurrence.person_id
AND cdm.observation.observation_date = cdm.visit_occurrence.visit_start_date
AND cdm.visit_occurrence.visit_type_concept_id = 32809;
```

OMOP Table name: fact_relationship

OMOP definition: 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).

Adaptation for ALPHA to OMOP mapping: This table connects two observations to from an episode. Within the **observation table**, between two observations, an episode is formed to describe the entry and exit.

Guidelines for Mapping ALPHA Residencies to OMOP fact_relationship

Destination Field	Field Description	Source Field	Logic	Comment
domain_concept_id_1	This field indicates which CDM domain the 1st record (fact) belongs to. (CONCEPT)		The concept of START TIME is taken here to show the start (entry event) of the episode domain_concept_id_1 = 4160040	Start time Domain ID: Observation Concept Class ID: Qualifier Value Vocabulary ID: SNOMED Concept ID: 4160040 Concept code: 398201009 Validity: Valid Concept: Standard Valid start: 31-Jul-2003

				Valid end: 31-Dec-2099
fact_id_1	This is not a concept, but a unique record ID in another CDM table from which the relationship is built	observation_id of first record, i.e., the start event of the episode	Get the start event id's observation_id from observation table	
domain_concept_id_2	This field indicates which CDM domain the 2nd record (fact) belongs to. (CONCEPT)		The concept of END is taken here to show the end (exit event) of the episode domain_concept_id_1 = 4129948	End Domain ID: Observation Concept Class ID: Qualifier Value Vocabulary ID: SNOMED Concept ID: 4129948 Concept code: 261782000 Validity: Valid Concept: Standard Valid start: 31-Jan-2002 Valid end: 31-Dec-2099
fact_id_2	This is not a concept, but a unique record ID in another CDM table from which the relationship is built	observation_id of second record, i.e., the end event of the episode	Get the end event id's observation_id from observation table	
relationship_concept_id	A concept which provides more context on how/why these two facts are connected/relate to each other		The concept of SINGLE EPISODE is taken here domain_concept_id_1 = 4041840	Single Episode Domain ID: Observation Concept Class ID: Qualifier Value Vocabulary ID: SNOMED Concept ID: 4041840 Concept code: 229803000 Validity: Valid Concept: Standard Valid start: 31-Jan-2002 Valid end: 31-Dec-2099

SQL Skeleton codes for ETL: ALPHA Residencies to OMOP fact_relationship

Following is a set of SQL skeleton code for reference purpose only for implementation of ETL to migrate data from ALPHA residencies to OMOP observation table. This SQL code is generated for PostgreSQL database where data is migrated from the INSPIRE database's alpha schema to cdm schema.

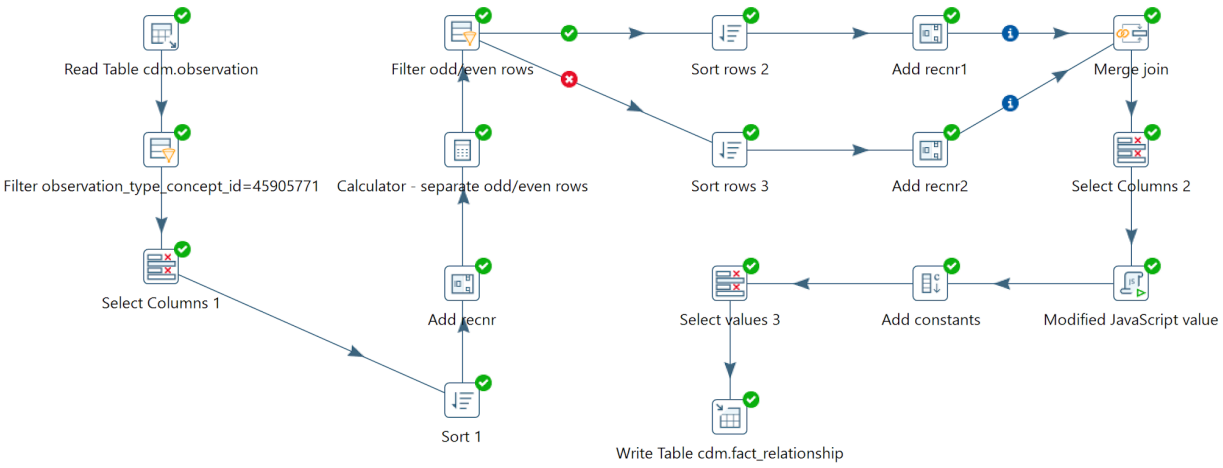
SQL statement to create the OMOP survey_conduct table in INSPIRE database's cdm schema

```
CREATE TABLE cdm.fact_relationship
(
 domain_concept_id_1 BIGINT
```

```
, fact_id_1 BIGINT
, domain_concept_id_2 BIGINT
, fact_id_2 BIGINT
, relationship_concept_id BIGINT
);
```

Implementation of ETL to populate fact_relatioship table

Here the implementation of ETL is showed using Pentaho Data Integration (Community Edition 9.1).



OMOP Table name: survey_conduct

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

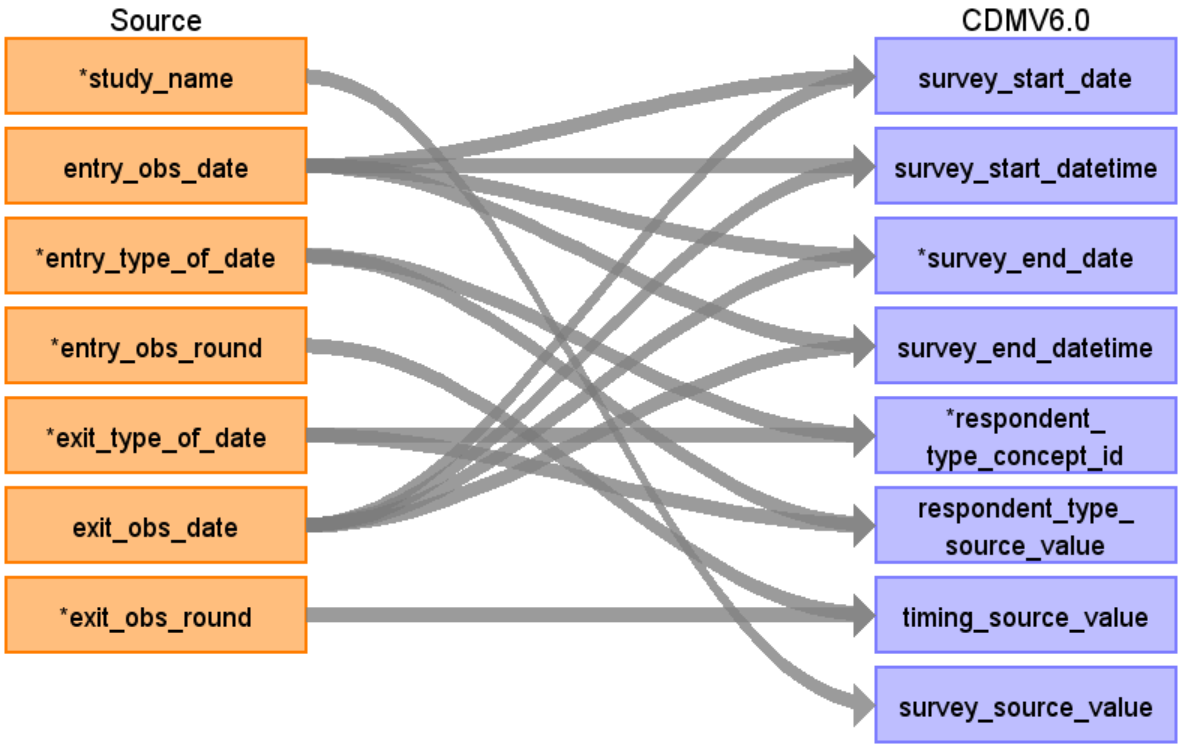
Adaptation for ALPHA to OMOP mapping: This table holds the details of the survey conducted i.e., the HDSS household visits for survey interviews. This table is very similar to the way the visit_detail and we have discussed to have chosen the survey_conduct table in this ALPHA to OMOP mapping. Here it is noted that for each visit, a set of survey questionnaire was

used for question/response and thus the visit_occurrence table stores the visit information and the survey_conduct stores the details of the visit.

Here the ALPHA residencies wide format (episode format) will get converted into long format (event format).

ALPHA Residencies to OMOP Survey_Conduct

Reading from residencies:



Guidelines for Mapping ALPHA Residencies to OMOP Survey_Conduct

Destination Field	Field Description	Source Field	Logic	Comment
survey_conduct_id	Unique identifier for each completed survey		Generate a sequential number to identify each survey conducted, i.e., the household visit details during HDSS survey	
person_id	A foreign key identifier to the Person in the PERSON table about whom the survey was completed.		This will be a foreign key identifier of the person for whom the survey was conducted.	Link this with person_id column of OMOP person table using person_source_value.
survey_concept_id	A foreign key to the predefined		This is a mandatory field, and having no concept id	This spec is a concept in a new VOCABULARY

	Concept identifier in the Standardized Vocabularies reflecting the name and identity of the survey		created yet, we are going to populate with value 0 for now, meaning 'No matching concept'	we are creating called Data Tabulation Models.
survey_start_date	Date on which the survey was started	entry_obs_date exit_obs_date	Map the entry_obs_date or exit_obs_date to populate this variable	Convert the wide format (episode format) to long format (event format). Store date in *YYYY-MM-DD* format
survey_start_datetime	Date and time the survey was started	entry_obs_date exit_obs_date	Map the entry_obs_date or exit_obs_date in TIMESTAMP format to populate this variable	Convert the wide format (episode format) to long format (event format). Store date in *YYYY-MM-DD 00:00:00* format
survey_end_date	Date on which the survey was completed	entry_obs_date exit_obs_date	Map the entry_obs_date or exit_obs_date to populate this variable. If the source field value is NULL then take a default date as '9999-12-31'.	This is a mandatory OMOP CDM field. Convert the wide format (episode format) to long format (event format) Store date in *YYYY-MM-DD* format
survey_end_datetime	Date and time the survey was completed	entry_obs_date exit_obs_date	Map the entry_obs_date or exit_obs_date in TIMESTAMP format to populate this variable	Convert the wide format (episode format) to long format (event format). Store date in *YYYY-MM-DD 00:00:00* format
provider_id	A foreign key to the provider in the provider table who was associated with the survey completion			Not applicable
assisted_concept_id	A foreign key to the predefined Concept identifier in the Standardized Vocabularies indicating whether the survey was completed with assistance		Not applicable, so populate it with value 0 as 'No matching concept'	DETAILS Domain ID: Metadata Concept Class: ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
respondent_type_concept_id	A foreign key to the predefined Concept identifier in the	entry_type_of_date	Map as follows:	

	Standardized Vocabularies reflecting the respondent type. Example: Research Associate, Patient	exit_type_of_ date	Reported by HH informant at interview (1) = Respondent (45882579) Reported by key informant (2) = Interview with informant (45881752) See SQL code below for the logic.	
timing_concept_id	A foreign key to the predefined Concept identifier in the Standardized Vocabularies that refers to a certain timing. Example: 3 month follow-up, 6 month follow-up		As of now, no matching concept id is found in the standard vocabulary and since it is a mandatory field, the timing_concept_id is populated with value 0 as 'No matching concept'	DETAILS Domain ID: Metadata Concept Class: ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
collection_method _concept_id	A foreign key to the predefined Concept identifier in the Standardized Vocabularies reflecting the data collection method (e.g. Paper, Telephone, Electronic Questionnaire)		Not applicable as ALPHA does not extract this information, so populate it with value 0 as 'No matching concept'	DETAILS Domain ID: Metadata Concept Class: ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099;
assisted_source_v alue	Source value representing whether patient required assistance to complete the survey. Example: Completed without assistance, Completed with assistance		ALPHA residencies do not store this detail	Not applicable
respondent_type_s ource_value	Source code representing role of person who completed the survey	entry_type_of _date exit_type_of_ date	Map the entry_type_of_date or exit_type_of_date to populate this variable. 1 = Reported by HH informant at interview 2 = Reported by key informant.	Map the entry_type or exit_type to populate this variable. Use the labels and not the numeric code.

			See SQL code below for the logic.	Convert the wide format (episode format) to long format (event format).
timing_source_value	Text string representing the timing of the survey. Example: Baseline, 6-month follow-up	entry_obs_round exit_obs_round	Map the entry_obs_round or exit_obs_round to populate this variable.	Convert the wide format (episode format) to long format (event format)
collection_method_source_value	The collection method as it appears in the source data		ALPHA residencies do not store this detail	Not applicable
survey_source_value	The survey name/title as it appears in the source data	study_name	Map it to the study_name from ALPHA residencies	
survey_source_concept_id	A foreign key to a predefined Concept that refers to the code for the survey name/title used in the source		Not applicable as ALPHA does not extract this information, so populate it with value 0 as 'No matching concept'	DETAILS Domain ID: Metadata Concept Class: ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
survey_source_identifier	Unique identifier for each completed survey in source system		ALPHA residencies do not store this detail	Not applicable
validated_survey_concept_id	A foreign key to the predefined Concept identifier in the Standardized Vocabularies reflecting the validation status of the survey		Not applicable, so populate it with value 0 as 'No matching concept'	DETAILS Domain ID: Metadata Concept Class: ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
validated_survey_source_value	Source value representing the validation status of the survey		ALPHA residencies do not store this detail	Not applicable
survey_version_number	Version number of the			Version number of questionnaires used for interview are not

	questionnaire or survey used			recorder in ALPHA data specs
visit_occurrence_id	A foreign key to the VISIT_OCCURRENCE table during which the survey was completed		Link it as foreign key to visit_occurrence_id form the visit_occurrence OMOP table	
response_visit_occurrence_id	A foreign key to the visit in the VISIT_OCCURRENCE table during which treatment was carried out that relates to this survey		Since HDSS does not provide any treatment and this field captures the treatment that was carried out related to this survey is not applicable.	

SQL Skeleton codes for ETL: ALPHA Residencies to OMOP Survey_Conduct

Following is a set of SQL skeleton code for reference purpose only for implementation of ETL to migrate data from ALPHA residencies to OMOP observation table. This SQL code is generated for PostgreSQL database where data is migrated from the INSPIRE database's alpha schema to cdm schema.

SQL statement to create the OMOP survey_conduct table in INSPIRE database's cdm schema

```
CREATE TABLE IF NOT EXISTS cdm.survey_conduct
(
  survey_conduct_id bigint NOT NULL,
  person_id bigint NOT NULL,
  survey_concept_id integer NOT NULL,
  survey_start_date date,
  survey_start_datetime timestamp without time zone,
  survey_end_date date NOT NULL,
  survey_end_datetime timestamp without time zone,
  provider_id bigint,
  assisted_concept_id integer NOT NULL,
  respondent_type_concept_id integer NOT NULL,
  timing_concept_id integer NOT NULL,
  collection_method_concept_id integer NOT NULL,
  assisted_source_value character varying(50),
  respondent_type_source_value character varying(100),
  timing_source_value character varying(100),
  collection_method_source_value character varying(100),
  survey_source_value character varying(100),
  survey_source_concept_id integer NOT NULL,
  survey_source_identifier character varying(100),
  validated_survey_concept_id integer NOT NULL,
  validated_survey_source_value character varying(100),
```

```
  survey_version_number character varying(20),
  visit_occurrence_id bigint,
  visit_detail_id bigint,
  response_visit_occurrence_id bigint,
  CONSTRAINT xpk_survey PRIMARY KEY (survey_conduct_id)
);
```

SQL statement to create a Sequence to generate the survey_conduct_id in PostgreSQL

```
CREATE SEQUENCE cdm.survey_conduct_id_seq;
```

SQL statement to insert data into cdm.survey_conduct table from alpha.residencies table

```
--Insert the start event details to convert the wide format to long format
INSERT INTO cdm.survey_conduct
(
  survey_conduct_id,
  person_id,
  survey_concept_id,
  survey_start_date,
  survey_start_datetime,
  survey_end_date,
  survey_end_datetime,
  provider_id,
  assisted_concept_id,
  respondent_type_concept_id,
  timing_concept_id,
  collection_method_concept_id,
  assisted_source_value,
  respondent_type_source_value,
  timing_source_value,
  collection_method_source_value,
  survey_source_value,
  survey_source_concept_id,
  survey_source_identifier,
  validated_survey_concept_id,
  validated_survey_source_value,
  survey_version_number,
  visit_occurrence_id,
  response_visit_occurrence_id
)
SELECT
  NEXTVAL('cdm.survey_conduct_id_seq')   AS survey_conduct_id,
  cdm.person.person_id AS person_id,
  0 AS survey_concept_id,
  residencies.entry_obs_date AS survey_start_date,
  residencies.entry_obs_date AS survey_start_datetime,
  CASE WHEN residencies.entry_obs_date IS NULL THEN '9999-12-31'
       ELSE residencies.entry_obs_date
```

```
    END AS survey_end_date,
    residencies.entry_obs_date AS survey_end_datetime,
    NULL AS provider_id,
    0 AS assisted_concept_id,
    CASE WHEN residencies.entry_type_of_date = 1 THEN 45882579
         WHEN residencies.entry_type_of_date = 2 THEN 45881752
         ELSE 0
     END AS respondent_type_concept_id,
    0 AS timing_concept_id,
    0 AS collection_method_concept_id,
    NULL AS assisted_source_value,
   CASE WHEN residencies.entry_type_of_date = 1 THEN '1 - Reported by HH informant at interview'
        WHEN residencies.entry_type_of_date = 2 THEN '2 - Reported by key informant'
          ELSE NULL
    END AS respondent_type_source_value,
    residencies.entry_obs_round AS timing_source_value,
    NULL AS collection_method_source_value,
    residencies.study_name AS survey_source_value,
    0 AS survey_source_concept_id,
    NULL AS survey_source_identifier,
    0 AS validated_survey_concept_id,
    NULL AS validated_survey_source_value,
    NULL AS survey_version_number,
    NULL AS visit_occurrence_id,
    NULL AS response_visit_occurrence_id
FROM alpha.residencies
INNER JOIN cdm.person
ON CAST(cdm.person.person_source_value AS INT) = CAST(alpha.residencies.idno AS INT);

--Insert the end event details to convert the wide format to long format
INSERT INTO cdm.survey_conduct
(
    survey_conduct_id,
    person_id,
    survey_concept_id,
    survey_start_date,
    survey_start_datetime,
    survey_end_date,
    survey_end_datetime,
    provider_id,
    assisted_concept_id,
    respondent_type_concept_id,
    timing_concept_id,
    collection_method_concept_id,
    assisted_source_value,
    respondent_type_source_value,
    timing_source_value,
    collection_method_source_value,
    survey_source_value,
    survey_source_concept_id,
```

```
  survey_source_identifier,
  validated_survey_concept_id,
  validated_survey_source_value,
  survey_version_number,
  visit_occurrence_id,
  response_visit_occurrence_id
)
SELECT
  NEXTVAL('cdm.survey_conduct_id_seq')  AS survey_conduct_id,
  cdm.person.person_id AS person_id,
  0 AS survey_concept_id,
  residencies.exit_obs_date AS survey_start_date,
  residencies.exit_obs_date AS survey_start_datetime,
  CASE WHEN residencies.exit_obs_date IS NULL THEN '9999-12-31'
          ELSE residencies.exit_obs_date
  END AS survey_end_date,
  residencies.exit_obs_date AS survey_end_datetime,
  NULL AS provider_id,
  0 AS assisted_concept_id,
 CASE WHEN residencies.exit_type_of_date = 1 THEN 45882579
       WHEN residencies.exit_type_of_date = 2 THEN 45881752
       ELSE 0
  END AS respondent_type_concept_id,
  0 AS timing_concept_id,
  0 AS collection_method_concept_id,
  NULL AS assisted_source_value,
  CASE WHEN residencies.exit_type_of_date = 1 THEN '1 - Reported by HH informant at interview'
        WHEN residencies.exit_type_of_date = 2 THEN '2 - Reported by key informant'
        ELSE NULL
  END AS respondent_type_source_value,
  residencies.exit_obs_round AS timing_source_value,
  NULL AS collection_method_source_value,
  residencies.study_name AS survey_source_value,
  0 AS survey_source_concept_id,
  NULL AS survey_source_identifier,
  0 AS validated_survey_concept_id,
  NULL AS validated_survey_source_value,
  NULL AS survey_version_number,
  NULL AS visit_occurrence_id,
  NULL AS response_visit_occurrence_id
FROM alpha.residencies
INNER JOIN cdm.person
ON CAST(cdm.person.person_source_value AS INT) = CAST(alpha.residencies.idno AS INT);
```

SQL statement to update the visit_occurrence_id by linking it to the visit_occurrence table

```
UPDATE cdm.survey_conduct
SET visit_occurrence_id = visit_occurrence.visit_occurrence_id
FROM cdm.visit_occurrence
```

WHERE cdm.survey_conduct.person_id = cdm.visit_occurrence.person_id
AND cdm.survey_conduct.survey_start_date = cdm.visit_occurrence.visit_start_date
AND cdm.visit_occurrence.visit_type_concept_id = 32809;

OMOP Table name: location_history

OMOP definition: The LOCATION HISTORY table stores relationships between Persons or Care Sites and geographic locations over time. **This table is new to CDM v6.0**.

Adaptation for ALPHA to OMOP mapping: The location_history table keeps a track of the movement of a person from one location to another. If the location changes to a new location in the next observation, that means there has been a movement that can be subsequently calculated as location exit and location entry for migrations. The importance of including this table in mapping from an LPS is to track the movement of persons.

The LOCATION table is not populated here because the information needed is not available in ALPHA data specifications except for the location id and thus will be of no additional value. This table linked with the person will give the location of the person at each event with start and end dates of the stay.

ALPHA Residencies to OMOP Location_History

Reading from residencies:



Guidelines for Mapping ALPHA Residencies to OMOP Location_History

Destination Field	Field Description	Source Field	Logic	Comment
location_id	This is the LOCATION_ID for the LOCATION_HISTORY record		Generate a sequential number to identify each location a person has stayed or lived in as per HDSS definitions.	Note: This location_id is not linked to the location table which has no signification in this LPS mapping
relationship_type_concept_id	This is the relationship between the location and the entity (PERSON, PROVIDER, or CARE_SITE)		Statically imputed, to the closest matching concept of 'living space' i.e., relationship_type_concept_id = 4049366.	DETAILS: Domain ID: Observation Concept Class ID: Observable Entity Vocabulary ID: SNOMED Concept ID: 4049366 Concept code: 20733006 Validity: Valid

				Concept: Standard Synonyms: Living place (observable entity) Valid start: 31-Jan-2002 Valid end: 31-Dec-2099
domain_id	The domain of the entity that is related to the location. Either PERSON, PROVIDER, or CARE_SITE		Statically imputed. The domain here is related to the PERSON so the domain_id will be 4129409	DETAILS Domain IDObservation Concept Class IDSocial Context Vocabulary ID SNOMED help_outline Concept ID4129409 Concept code125676002 ValidityValid ConceptStandard Synonyms Person (person) Valid start31-Jan-2002 Valid end31-Dec-2099
entity_id	The unique identifier for the entity. References either person_id, provider_id, or care_site_id, depending on domain_id		Map it to person_id from OMOP person table	This is a mandatory field. The person_id will link this table to the OMOP person table
start_date	The date the relationship started	entry_date	Map the location (hhold_id) entry date. If the source field value is NULL then take a default date as '9999-12-31'.	This is a mandatory field. Store the date in *YYYY-MM-DD* format
end_date	The date the relationship ended	exit_date	Map the location (hhold_id) exit date	Store the date in *YYYY-MM-DD* format

SQL Skeleton codes for ETL: ALPHA Residencies to OMOP Location_History

Following is a set of SQL skeleton code for reference purpose only for implementation of ETL to migrate data from ALPHA residencies to OMOP observation table. This SQL code is generated for PostgreSQL database where data is migrated from the INSPIRE database's alpha schema to cdm schema.

SQL statement to create the OMOP location_history table in INSPIRE database's cdm schema

```
CREATE TABLE IF NOT EXISTS cdm.location_history
(
  location_id bigint NOT NULL,
```

```
  location_id bigint NOT NULL,
  relationship_type_concept_id integer NOT NULL,
  domain_id character varying(50),
  entity_id bigint NOT NULL,
  start_date date NOT NULL,
  end_date date,
  CONSTRAINT xpk_location_history PRIMARY KEY (location_history_id)
);
```

SQL statement to create a Sequence to generate the visit_location_id in PostgreSQL

```
CREATE SEQUENCE cdm.location_id_seq;
```

SQL statement to insert data into cdm.location table from alpha.residencies table

```
INSERT INTO cdm.location_history
(
  location_id,
  relationship_type_concept_id,
  domain_id,
  entity_id,
  start_date,
  end_date
)
SELECT
  NEXTVAL('cdm.location_id_seq')  AS location_id,
  4049366 AS relationship_type_concept_id,
  4129409 AS domain_id,
  cdm.person.person_id AS entity_id,
  CASE WHEN residencies.entry_date IS NULL THEN '9999-12-31'
       ELSE residencies.entry_date
  END AS start_date,
  residencies.exit_date AS end_date
FROM alpha.residencies
INNER JOIN cdm.person
ON CAST(cdm.person.person_source_value AS INT) = CAST(alpha.residencies.idno AS INT);
```

Mapping ALPHA HIV Test Data to OMOP v6.0 CDM

In this section of the document, we will discuss the mapping process from ALPHA HIV testing (ALPHA Data Spec 6.2b) to OMOP v6.0 CDM.

Source Table(s) Structure:

Table: HIV Test data

HIV test data is for the results of HIV tests conducted during sero-surveys, population-based studies or from clinic data.

The following table shows the ALPHA HIV tests data specification field names with descriptions.

Field	Description	Type	Mapping
idno	Person ID number - Numeric IDs long integer format, unique for an individual	integer	
study_name	Name of the study field site.	character varying	
test_report_date	Date of HIV test report.	date	
hiv_test_date	Date of HIV test. If test carried out in survey or study clinic date will be known exactly, if retrospectively reported by respondent may be approximated to mid-month or mid-year	date	
hiv_test_result	HIV test result. 0 - negative 1 - positive 2 - indeterminate 3 - not reported Indeterminate means test was part of study, but results were inconclusive; not reported means that participant said they had an HIV test outside of study setting but did not disclose result in interview. If participant says they do not know test result, code this as not reported, but only if your research study has no record of the result. If result is recorded in research study or clinic data base do not use not reported code.	smallint	
informed_of_result	Whether or not the participant was informed of test result. 0 - no 1 - yes 8 - Don't know/not asked no codes typical for anonymised tests in sero-surveys; yes, codes typical for VCT or PICT. It is possible for participant to say they were informed of result, even if they did not want to report it in survey interview.	smallint	
source_of_test_information	Whether the test information comes from your research study, or part of a linked clinical record or self-reported by respondent. 1 - part of a population-based study 2 - part of a special research study 3 - clinical record- HIV clinic 4 - self-reported by respondent 5 - report by proxy respondent at VA 6 - clinical record- walk in VCT 7 - clinical record- PMTCT/ANC	smallint	

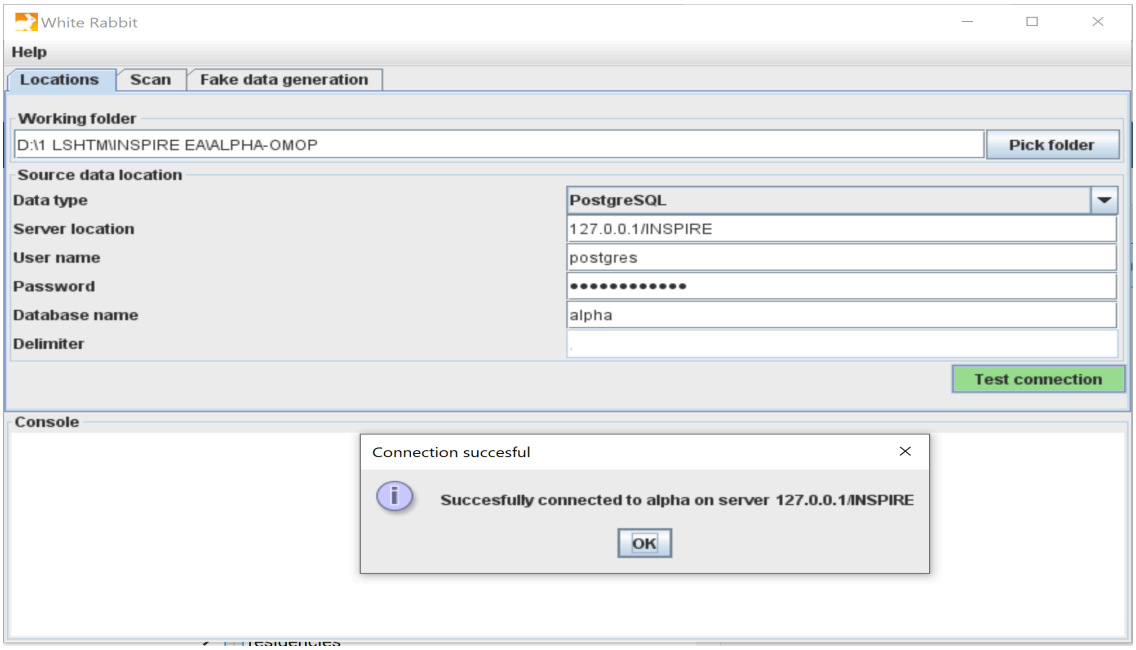
	8 - clinical record- other to distinguish between routine tests in surveys, extra tests in study clinics with results directly recorded in study data base, self-disclosed results and results from VA proxy respondents		
test_assumption	Most sites will code all tests as "0" for this variable. Sites that do not test, in their population-based study, someone who they have recently tested as part of a special study should code this variable as "1". Those studies that do not test repeat positives should use code "2" here for people who were not tested because they were already tested and found positive in the past. It is essential that these codes are only used on people who participated in the study and did not refuse to test. Use them only for people who were not tested simply due to study protocol, as outlined above. 0 - New test 1 - Test from previous study used 2 - Previous positive HIV test used 3 - Self reported instead of testing (Kisumu)	smallint	
original_hiv_test_result	This is the HIV test result before any changes were made for example the final test result "hiv_test_result" may be missing but was originally reported as negative. This may arise due to site specific discussions about retro converters and how to deal with them. It should usually be identical to "hiv_test_result". 0 negative 1 positive 2 indeterminate 3 not reported	smallint	
survey_round_name	If test was part of community survey name of round in which the data were collected. If the testing was done as part of a survey this variable should denote the name of the survey for example "sero1" or "TBSurvey2" etc it should be in string format.	character varying	

Steps to scan the ALPHA HIV Test table using WhiteRabbit

The following steps were performed to generate the scan report and fake data from ALPHA HIV Test:

Note here that as part of the data preparation process for this documentation, a dummy ALPHA dataset was migrated to a PostgreSQL database schema.

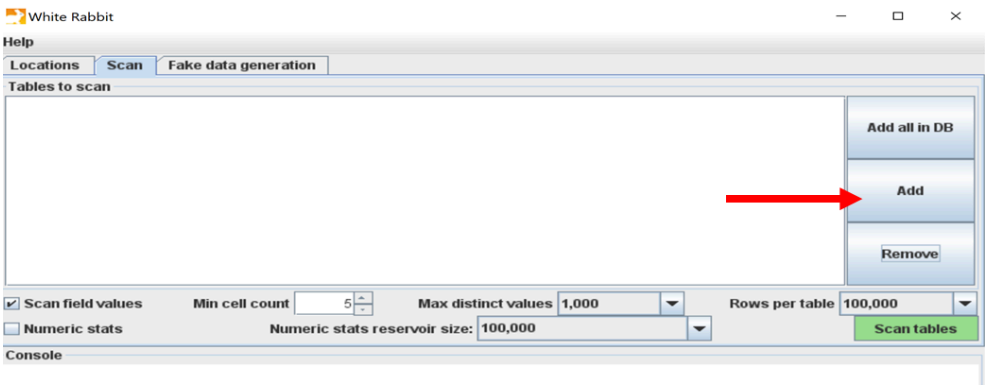
(1) Launched WhiteRabbit
(2) In the Location tab
    a. Picked the working folder.
    b. Connected to the PostgreSQL database
    c. Tested the connection



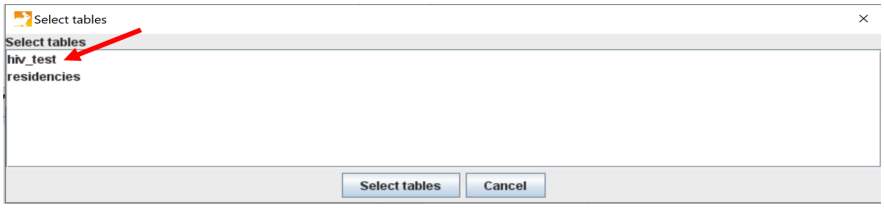
(3) In the Scan tab

a. Added the HIV Test table
b. Checked the Scan field values and Numeric stats checkbox.
c. Kept the other fields to default values and then clicked on scan button

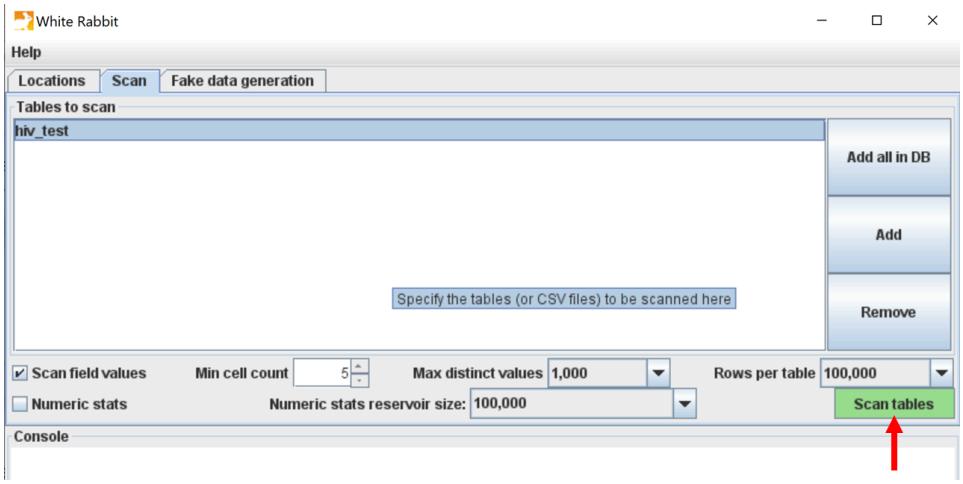
(A) Add



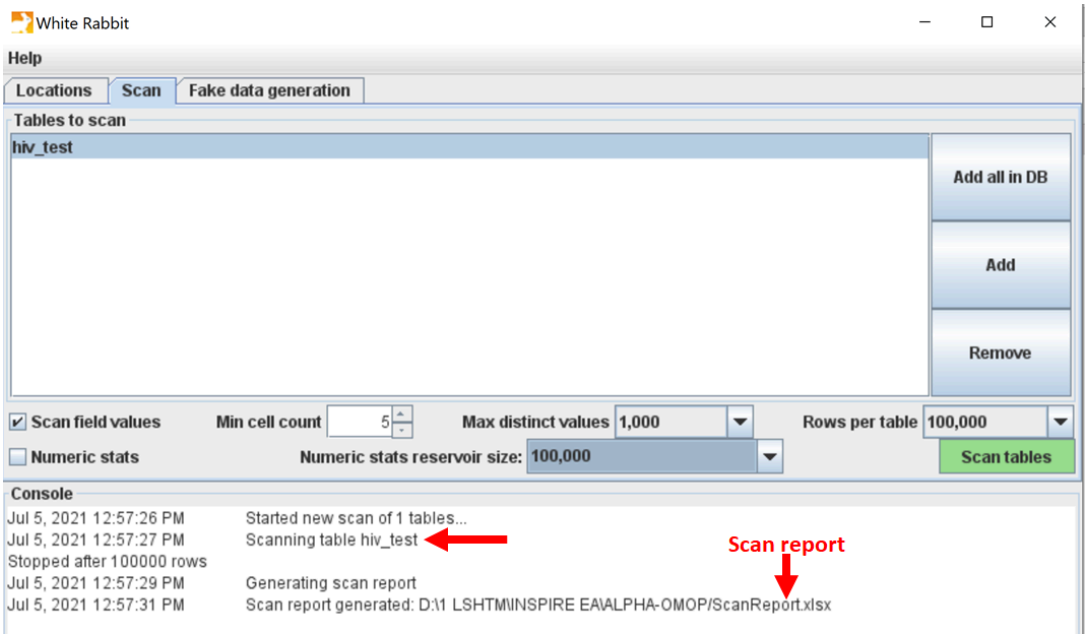
(B) Select the table



(C) Scan the table



(D) The scan report generated



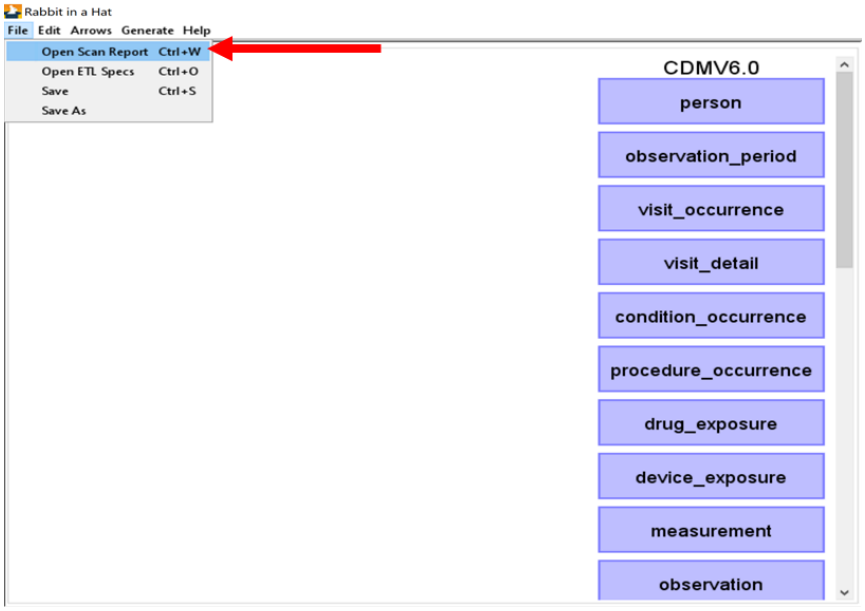
(E) The sample scan report looked like this.

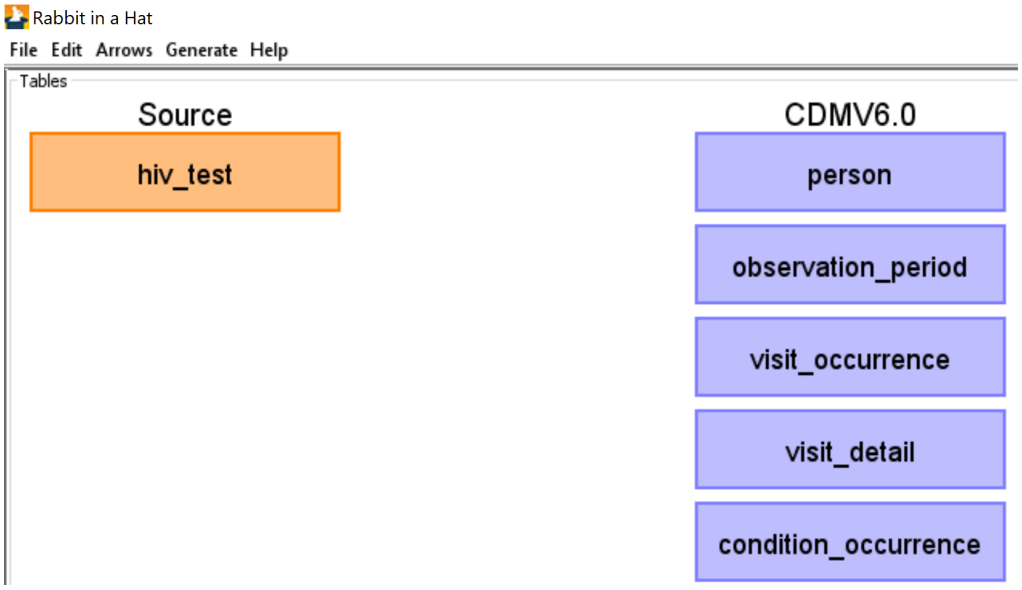
	A	B	C	D	E	F	G	H	I	J
1	Table	Field	Description	Type	Max length	N rows	N rows checked	Fraction empty	N unique values	Fraction unique
2	hiv_test	idno		character varying	8	198491	100000	0.0%	36613	36.6%
3	hiv_test	study_name		character varying	6	198491	100000	0.0%	1	0.0%
4	hiv_test	test_report_date		date	0	198491	100000	100.0%	1	0.0%
5	hiv_test	hiv_test_date		date	10	198491	100000	0.5%	6674	6.7%
6	hiv_test	hiv_test_result		smallint	1	198491	100000	1.1%	5	0.0%
7	hiv_test	informed_of_result		smallint	1	198491	100000	0.0%	1	0.0%
8	hiv_test	source_of_test_information		smallint	1	198491	100000	0.0%	1	0.0%
9	hiv_test	test_assumption		bigint	1	198491	100000	0.0%	1	0.0%
10	hiv_test	original_hiv_test_result		smallint	0	198491	100000	100.0%	1	0.0%
11	hiv_test	survey_round_name		smallint	2	198491	100000	0.0%	25	0.0%

Steps to start the mapping process of ALPHA HIV Test to OMOP using Rabbit In A Hat

The following steps were performed to generate the ETL process documentation for data mapping from ALPHA Residencies to OMOP:

(1) Opened the Scan Report
   a. Clicked on File from the menu bar
   b. Clicked on Open Scan Report (Shortcut Ctrl+w)
   c. Looked into the scan report folder and opened the scan report





Source Data Mapping Approach to CDMV6.0



The ALPHA HIV test, which stores the HIV test results for the population is been mapped to two OMOP CDM tables as show in the diagram above.

OMOP Table name: visit_occurrence

OMOP Definition: 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.

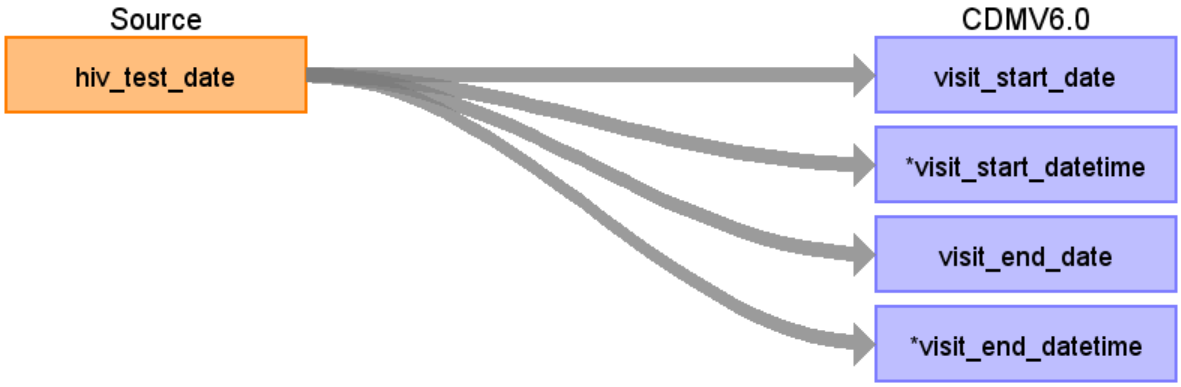
Adaptation for ALPHA to OMOP mapping: This OMOP CDM table will contain the visit date on which the HIV test was done. The assumption here is that the HIV test date would be same as the clinic visit date.

ALPHA HIV Test to OMOP visit_occurrence

Reading from HIV test: The ALPHA HIV test details stores the HIV test date and that date is taken as the clinic visit date for each person in visit_occurrence table. For persons who were tested HIV positive been extracted and migrated. It may happen that a person has been tested multiple times at different dates and all these dates were taken as clinic visit dates.

ALPHA HIV Test to OMOP visit_occurrence

Reading from HIV test: Here the test dates will be migrated to the visit occurrences.



Guidelines for Mapping ALPHA HIV Test to OMOP Visit_Occurrence

Destination Field	Field Description	Source Field	Logic	Comment
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.		Generate a sequential number to identify each visit occurrence uniquely, i.e., assigning a unique number to each HIV test dates migrated as clinic visit.	
person_id			This will be a foreign key identifier of the person for whom the visit occurrence is defined.	Link this with person_id column of OMOP person table using person_source_value.
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.		Statically imputed. The visit_concept_id to be used is “Clinic” visit_concept_id = 4119839.	DETAILS: Domain ID: Visit Concept Class ID: Location Vocabulary ID: SNOMED Concept ID: 4119839 Concept code: 257585005 Validity: Valid Concept: Non-standard Synonyms: Clinic (environment) Valid start: 31-Jan-2002

				Valid end: 31-Dec-2099
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.	hiv_test_date	Map the hiv_test_date to get the clinic visit date	Store date in YYYY-MM-DD format.
visit_start_datetime	If no time is given for the start date of a visit, set it to midnight (00:00:0000).	hiv_test_date	Map the hiv_test_date to get the clinic visit date in TIMESTAMP format. If hiv_test_date is NULL in source value, replace it with 9999-12-31.	This is a mandatory field. Store date in YYYY-MM-DD 00:00:00 format.
visit_end_date	For inpatient visits the end date is typically the discharge date.	hiv_test_date	Map the hiv_test_date to get the clinic visit date	Store date in YYYY-MM-DD format.
visit_end_datetime	If no time is given for the end date of a visit, set it to midnight (00:00:0000).	hiv_test_date	Map the hiv_test_date to get the clinic visit date in	This is a mandatory field. Store date in YYYY-MM-DD 00:00:00 format.
visit_type_concept_id	Use this field to understand the provenance of the visit record, or where the record comes from.		Statically imputed. The visit_concept_id to be used is “Clinical Study visit” visit_type_concept_id = 44818519.	DETAILS: Domain ID: Type Concept Concept Class ID: Visit Type Vocabulary ID: Visit Type Concept ID: 44818519 Concept code: OMOP4822463 Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
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 source, this can be reflected in the event tables			Not applicable
care_site_id	This field provides information about the care site where the visit took place.			Not applicable
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.)			Not applicable

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. If not available set to 0.		Not applicable as there is no visit_source_value, so populate visit_source_concept_id with value 0 as 'No matching concept'	Mandatory field. DETAILS: Domain ID: Metadata Concept Class ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
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.		Not applicable as there is no admitting_source_value, so populate admitting_source_concept_id with value 0 as 'No matching concept'	Mandatory field. DETAILS: Domain ID: Metadata Concept Class ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
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.			Not applicable
discharge_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 discharged to home or sent to a long-term care facility, for example.		Not applicable as there is no discharge_to_source_value, so populate discharge_to_concept_id with value 0 as 'No matching concept'	Mandatory field. DETAILS: Domain ID: Metadata Concept Class ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
discharge_to_source_value	This information may be called something different in the source data but the field is			Not applicable

	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.			
preceding_visit_occurrence_id	Use this field to find the visit that occurred for the person prior to the given visit. There could be a few days or a few years in between.			Not applicable

SQL Skeleton codes for ETL: ALPHA HIV Test to OMOP Visit_Occurrence

Following is a set of SQL skeleton code for reference purpose only for implementation of ETL to migrate data from ALPHA HIV Test to OMOP visit_occurrence table. This SQL code is generated for PostgreSQL database where data is migrated from the INSPIRE database's alpha schema to cdm schema.

SQL statement to create the OMOP visit_occurrence table in INSPIRE database's cdm schema

Table already created in the residencies ETL process, so no need to create here. Data will get appended into this table.

SQL statement to create a Sequence to generate the visit_occurrence_id in PostgreSQL

The visit_occurrence_id sequence is already created in the residences ETL process, so no need to create here. The same e sequence will be used to generate the visit_occurence_id in this ETL process.

SQL statement to insert data into cdm.visit_occurrence table from alpha.hiv_test table

The following SQL statement will append data into the visit_occurence table.

```
INSERT INTO cdm.visit_occurrence
(
  visit_occurrence_id,
  person_id,
  visit_concept_id,
  visit_start_date,
  visit_start_datetime,
  visit_end_date,
  visit_end_datetime,
  visit_type_concept_id,
  provider_id,
```

```
    care_site_id,
    visit_source_value,
    visit_source_concept_id,
    admitting_source_concept_id,
    admitting_source_value,
    discharge_to_concept_id,
    discharge_to_source_value,
    preceding_visit_occurrence_id
)
SELECT
        NEXTVAL('cdm.visit_occurrence_id_seq')  AS visit_occurrence_id,
    cdm.person.person_id AS person_id,
    4119839 AS visit_concept_id,
    hiv_test.hiv_test_date AS visit_start_date,
        CASE WHEN hiv_test.hiv_test_date IS NULL THEN '9999-12-31'
            ELSE hiv_test.hiv_test_date
        END AS visit_start_datetime,
    hiv_test.hiv_test_date AS visit_end_date,
        CASE WHEN hiv_test.hiv_test_date IS NULL THEN '9999-12-31'
            ELSE hiv_test.hiv_test_date
        END AS visit_end_datetime,
    44818519 AS visit_type_concept_id,
    NULL AS provider_id,
    NULL AS care_site_id,
    NULL AS visit_source_value,
    0 AS visit_source_concept_id,
    0 AS admitting_source_concept_id,
    NULL AS admitting_source_value,
    0 AS discharge_to_concept_id,
    NULL AS discharge_to_source_value,
    NULL AS preceding_visit_occurrence_id
FROM alpha.hiv_test
INNER JOIN cdm.person
ON CAST(cdm.person.person_source_value AS INT) = CAST(alpha.hiv_test.idno AS INT);
```

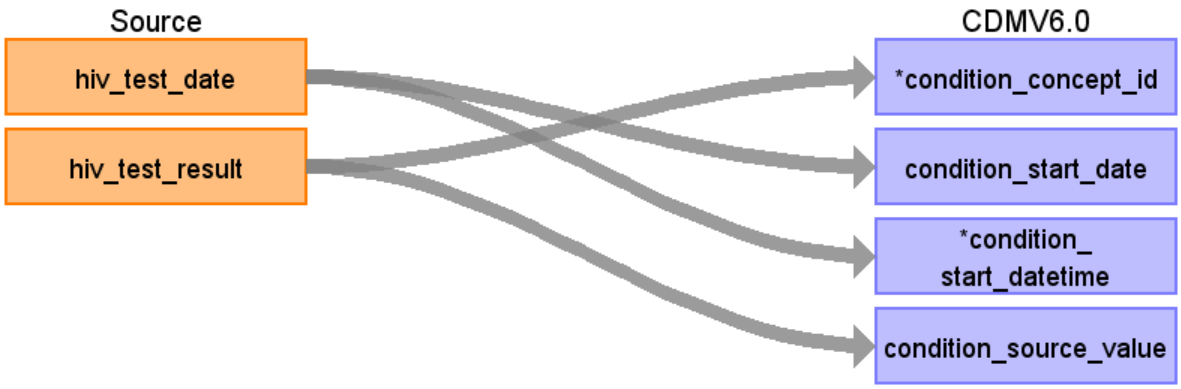
OMOP Table name: condition_occurrence

OMOP Definition: 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.

Adaptation for ALPHA to OMOP mapping: This OMOP CDM table will contain the condition of a person whose HIV test records are available in the ALPHA HIV Test table.

ALPHA HIV Test to OMOP condition_occurrence

Reading from HIV test: The ALPHA HIV test details, i.e., the event that recorded the HIV test result has been mapped to this table. It may happen that a person has been tested multiple times at different dates and all these test results were migrated here.



Guidelines for Mapping ALPHA HIV Test to OMOP Condition_Occurrence

Destination Field	Field Description	Source Field	Logic	Comment
condition_occurrence_id	The unique key given to a condition record for a person. 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 individual, unique		Generate a sequential number to identify each occurrence of the condition recorded for a person to be HIV positive.	
person_id	The PERSON_ID of the PERSON for whom the condition is recorded.		This will be a foreign key identifier of the person for whom the survey was conducted.	Link this with person_id column of OMOP person table using person_source_value.
condition_concept_id	This 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	hiv_test_result	Statically imputed. Source value is 0 (HIV Negative) then condition_concept_id = 4013105 Source value is 1 (HIV Positive) then condition_concept_id = 4013106	DETAILS: Domain ID: Condition Concept Class ID: Clinical Finding Vocabulary ID: SNOMED Concept ID: 4013106 Concept code: 165816005 Validity: Valid Concept: Standard Valid start: 31-Jan-2002

			Source value is 2 (HIV Indertminate) then condition_concept_id = 4088484	Valid end: 31-Dec-2099 Domain ID: Measurement Concept Class ID: Clinical Finding Vocabulary ID: SNOMED Concept ID: 4013105 Concept code: 165815009 Validity: Valid Concept: Standard Valid start: 31-Jan-2002 Valid end: 31-Dec-2099 Domain ID: Observation Concept Class ID: Qualifier Value Vocabulary ID: SNOMED Concept ID: 4088484 Concept code: 280416009 Validity: Valid Concept: Standard Synonyms Valid start: 31-Jan-2002 Valid end: 31-Dec-2099
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	hiv_test_date	The start date of the HIV positive condition is not recorded and thus the HIV test date which gave a HIV positive result is taken as the condition start date.	
condition_start_datetime	If a source does not specify datetime the convention is to set the time to midnight (00:00:0000)	hiv_test_date	The start date of the HIV positive condition is not recorded and thus the HIV test date which gave a HIV positive result is taken as the condition start date.	
condition_end_date	Use this date to determine the end date of the condition. Most often data			Not recorded and thus not applicable

	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.			
condition_end_datetime	If a source does not specify datetime the convention is to set the time to midnight (00:00:0000)			Not recorded and thus not applicable
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.		Statically imputed. The condition_type_concept_id to type concept “Case Report Form” condition_type_concept_id = 32809	DETAILS: Domain ID: Type Concept Concept Class ID: Type Concept Vocabulary ID: Type Concept Concept ID: 32809 Concept code: OMOP4976882 Validity: Valid Concept: Standard Valid start: 20-Aug-2020 Valid end: 31-Dec-2099
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 laboratory findings, if the diagnosis was exclusionary, or if it was a preliminary diagnosis, among others.		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. If not available, set to 0. No matching condition status concept id is found, so populate it with value 0 as ‘No matching concept’	DETAILS Domain ID: Metadata Concept Class: ID: Undefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099
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			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.

	that the condition is no longer occurring.			
provider_id	he provider associated with condition record, e.g. the provider who made the diagnosis or the provider who recorded the symptom			Not applicable
visit_occurrence_id	The visit during which the condition occurred.	visit_occurrence_id .visit_occurrence	FK. Link to visit_occurrence_id from visit_occurrence table.	The HIV test date is assumed to be the visit occurrence and thus the id generated for that visit date is linked here.
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.			The information on visit or interview that recorded the HIV test data is not stored in ALPHA HIV Test data specs and thus cannot be tracked to any visit details and thus left blank here
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.	hiv_test_result	HIV test result. 0 - negative 1 - positive 2 - indeterminate 3 - not reported	Only positive HIV test result are stored here.
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		No matching condition source concept id is found is recorded, so populate it with value 0 as 'No matching concept'	DETAILS Domain ID: Metadata Concept Class: IDUndefined Vocabulary ID: None Concept ID: 0 Concept code: No matching concept Validity: Valid Concept: Non-standard Valid start: 01-Jan-1970 Valid end: 31-Dec-2099

	adequately represent the source detail for the Condition necessary for a given analytic use case. Consider using CONDITION_CONCEPT_ID 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.			Not applicable

SQL Skeleton codes for ETL: ALPHA HIV Test to OMOP Condition_Occurrence

Following is a set of SQL skeleton code for reference purpose only for implementation of ETL to migrate data from ALPHA HIV Test to OMOP condition_occurrence table. This SQL code is generated for PostgreSQL database where data is migrated from the INSPIRE database's alpha schema to cdm schema.

SQL statement to create the OMOP condition_occurrence table in INSPIRE database's cdm schema.

```
CREATE TABLE IF NOT EXISTS cdm.condition_occurrence
(
  condition_occurrence_id bigint NOT NULL,
  person_id bigint NOT NULL,
  condition_concept_id integer NOT NULL,
  condition_start_date date,
  condition_start_datetime timestamp without time zone NOT NULL,
  condition_end_date date,
  condition_end_datetime timestamp without time zone,
  condition_type_concept_id integer NOT NULL,
  condition_status_concept_id integer NOT NULL,
  stop_reason character varying(20),
  provider_id bigint,
  visit_occurrence_id bigint,
  visit_detail_id bigint,
  condition_source_value character varying(50),
  condition_source_concept_id integer NOT NULL,
  condition_status_source_value character varying(50),
  CONSTRAINT xpk_condition_occurrence PRIMARY KEY (condition_occurrence_id)
);
```

SQL statement to create a Sequence to generate the condition_occurrence_id in PostgreSQL

```
CREATE SEQUENCE cdm.condition_occurrence_id_seq;
```

SQL statement to insert data into cdm.condition_occurrence table from alpha.hiv_test table

```
INSERT INTO cdm.condition_occurrence
(
  condition_occurrence_id,
  person_id,
  condition_concept_id,
  condition_start_date,
  condition_start_datetime,
  condition_end_date,
  condition_end_datetime,
  condition_type_concept_id,
  condition_status_concept_id,
  stop_reason,
  provider_id,
  visit_occurrence_id,
  visit_detail_id,
  condition_source_value,
  condition_source_concept_id,
  condition_status_source_value
)
SELECT
          NEXTVAL('cdm.condition_occurrence_id_seq')  AS condition_occurrence_id,
  cdm.person.person_id AS person_id,
  CASE WHEN hiv_test.hiv_test_result = 0 THEN 4013105
       WHEN hiv_test.hiv_test_result = 1 THEN 4013106
       WHEN hiv_test.hiv_test_result = 2 THEN 4088484
       ELSE 0
  END AS condition_concept_id,
 hiv_test.hiv_test_date AS condition_start_date,
 CASE WHEN hiv_test.hiv_test_date IS NULL THEN '9999-12-31'
      ELSE hiv_test.hiv_test_date
 END AS condition_start_datetime,
  NULL AS condition_end_date,
  NULL AS condition_end_datetime,
  32809 AS condition_type_concept_id,
  0 AS condition_status_concept_id,
  NULL AS stop_reason,
  NULL AS provider_id,
  NULL AS visit_occurrence_id,
  NULL AS visit_detail_id,
  hiv_test.hiv_test_result AS condition_source_value,
  0 AS condition_source_concept_id,
  NULL AS condition_status_source_value
FROM alpha.hiv_test
```

INNER JOIN cdm.person
ON CAST(cdm.person.person_source_value AS INT) = CAST(alpha.hiv_test.idno AS INT);

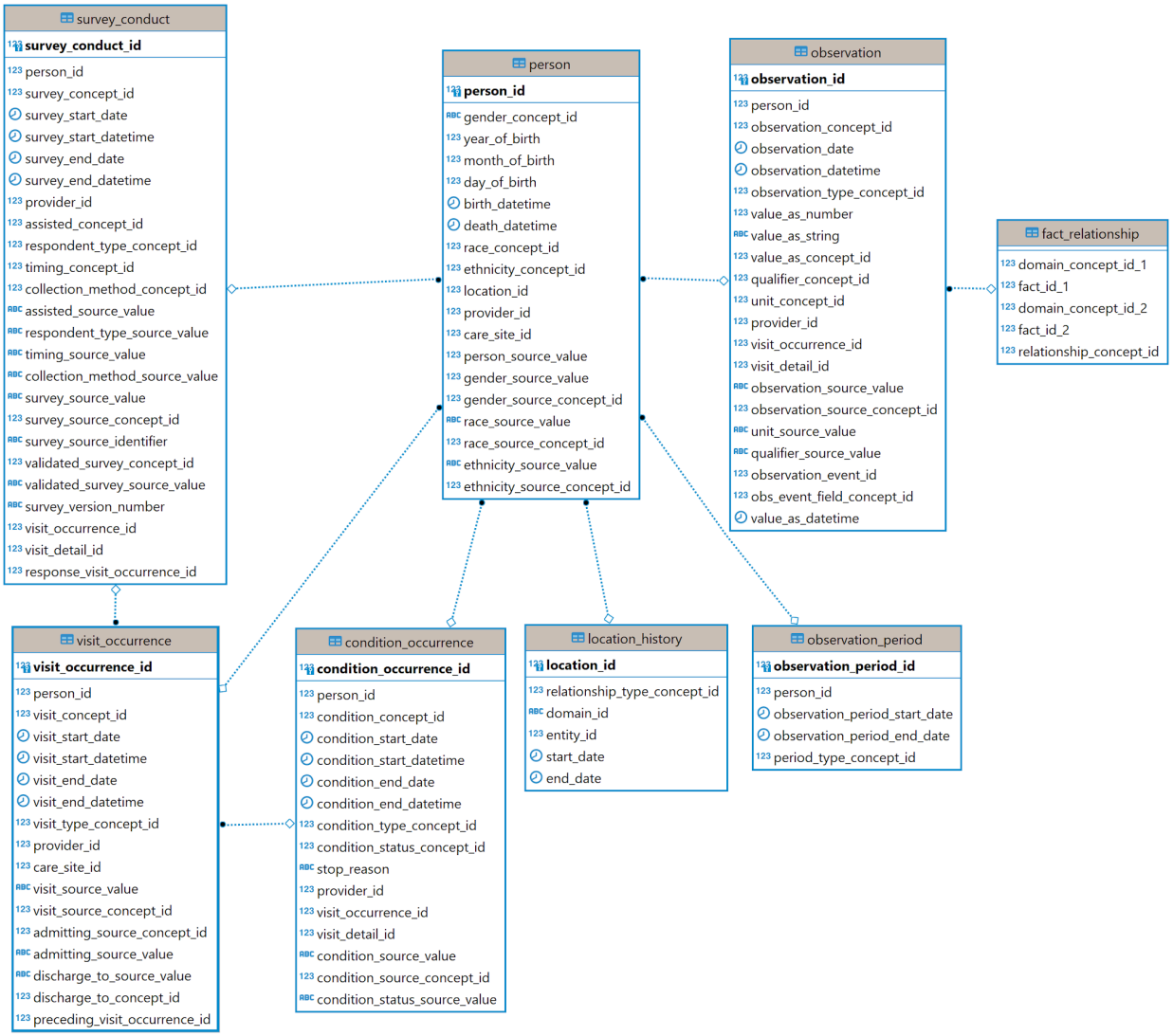
SQL statement to update the visit_occurrence_id by linking it to the visit_occurrence table

UPDATE cdm.condition_occurrence
SET visit_occurrence_id = cdm.visit_occurrence.visit_occurrence_id
FROM cdm.visit_occurrence
WHERE cdm.condition_occurrence.person_id = cdm.visit_occurrence.person_id
AND cdm.condition_occurrence.condition_start_datetime = cdm.visit_occurrence.visit_start_datetime
AND cdm.visit_occurrence.visit_type_concept_id = 44818519;

Entity Relation Diagram of OMOP CDM Tables after ALPHA Residencies and HIV Tests Data mapping

This entity relation diagram has been created using an open-source tool named DBeaver.

URL: https://dbeaver.io/


survey_conduct
survey_conduct_id
person_id
survey_concept_id
survey_start_date
survey_start_datetime
survey_end_date
survey_end_datetime
provider_id
assisted_concept_id
respondent_type_concept_id
timing_concept_id
collection_method_concept_id
assisted_source_value
respondent_type_source_value
timing_source_value
collection_method_source_value
survey_source_value
survey_source_concept_id
survey_source_identifier
validated_survey_concept_id
validated_survey_source_value
survey_version_number
visit_occurrence_id
visit_detail_id
response_visit_occurrence_id
person
person_id
gender_concept_id
year_of_birth
month_of_birth
day_of_birth
birth_datetime
death_datetime
race_concept_id
ethnicity_concept_id
location_id
provider_id
care_site_id
person_source_value
gender_source_value
gender_source_concept_id
race_source_value
race_source_concept_id
ethnicity_source_value
ethnicity_source_concept_id
observation
observation_id
person_id
observation_concept_id
observation_date
observation_datetime
observation_type_concept_id
value_as_number
value_as_string
value_as_concept_id
qualifier_concept_id
unit_concept_id
provider_id
visit_occurrence_id
visit_detail_id
observation_source_value
observation_source_concept_id
unit_source_value
qualifier_source_value
observation_event_id
obs_event_field_concept_id
value_as_datetime
fact_relationship
domain_concept_id_1
fact_id_1
domain_concept_id_2
fact_id_2
relationship_concept_id
visit_occurrence
visit_occurrence_id
person_id
visit_concept_id
visit_start_date
visit_start_datetime
visit_end_date
visit_end_datetime
visit_type_concept_id
provider_id
care_site_id
visit_source_value
visit_source_concept_id
admitting_source_concept_id
admitting_source_value
discharge_to_source_value
discharge_to_concept_id
preceding_visit_occurrence_id
condition_occurrence
condition_occurrence_id
person_id
condition_concept_id
condition_start_date
condition_start_datetime
condition_end_date
condition_end_datetime
condition_type_concept_id
condition_status_concept_id
stop_reason
provider_id
visit_occurrence_id
visit_detail_id
condition_source_value
condition_source_concept_id
condition_status_source_value
location_history
location_id
relationship_type_concept_id
domain_id
entity_id
start_date
end_date
observation_period
observation_period_id
person_id
observation_period_start_date
observation_period_end_date
period_type_concept_id

LPS best practices

Longitudinal population studies (LPS) extend and subsume two OMOP CDM domains – PERSON and VISIT_OCCURRENCEs. In LPS, PERSON figures into a DEMOGRAPHICS domain and VISIT_OCCURRENCES (along with VISIT_DETAIL) figures into the PROTOCOL domain.

Some demographics domain best practices

Most generally:

- An OBSERVATION_PERIOD is relative to an HDSS. It captures HDSS in-migrations (including birth) and out-migrations (including death). Internal migration within an HDSS is captured by LOCATION_HISTORY. If a PERSON leaves an HDSS and subsequently returns, s/he will have multiple OBSERVATION_PERIODs.
- A VISIT_OCCURRENCE holds the visits that were made to households as part of the HDSS longitudinal population study
- VISIT_DETAIL isn't used here. That being said, VISIT_DETAIL comes into play in an LPS in which during the same visit multiple surveys and/or labs are administered.
- Here SURVEY_CONDUCT augments a VISIT_OCCURRENCE with certain specifics like the survey / data tabulation model (CRF) concept, the periodicity of each survey / CRF and the round. Data tabulation model is its own vocabulary. survey_concept_id refers to concepts from this vocabulary or, alternatively, scales, questionnaires, inventories, etc. mostly from SNOMED. And periodicity has a category set in the LPS de novo vocabulary. Additionally, survey_source_value refers to the instrument on that mapped to the data exchange/tabulation model so we can trace SURVEY_CONDUCT back in time first to an exchange/tabulation model and then to an instrument if applicable.
- Each SURVEY_CONDUCT includes a foreign key to a VISIT_OCCURRENCE. This is a necessary one-to-one relationship. This relationship is implemented using the SURVEY_CONDUCT visit_occurrence_id.
- OBSERVATION captures migration (including birth and death) events. If these events are either in-migrations or out-migrations relative to an HDSS, then these OBSERVATIONs use observation_event_id and obs_event_field_concept_id to fashion a foreign key where observation_event_id points to a record in the OBSERVATION_PERIOD table and obs_event_field_concept_id points to the table itself. Likewise, if these events are internal migration events, then observation_event_id points to a record in the LOCATION_HISTORY table and obs_event_field_concept_id points to the table itself.

More specifically:

- **LOCATION_HISTORY.relationship_type_concept_id** is the relationship between the location and the entity. When the entity is a PERSON, if we want to indicate that the

location is the person's residence, the convention we will follow is that **relationship_type_concept_id** will take SNOMED's "living place". Note that "living place" is just one of many concepts that are in a relationship with "residence and accommodation circumstances". In future specs when we are capturing other residence and accommodation circumstances, as a rule we will use one of the "residence and accommodation circumstances" related concepts. In this way we are fashioning a category set in the Standardized Vocabulary in which "residence and accommodation circumstances" the is the variable concept which is associated with multiple allowable category concepts through an "Interprets of (SNOMED)" relationship.

Some protocol domain best practices

Use SURVEY_CONDUCT.survey_concept_id to refer to the actual instrument mapped to a data tabulation model. This might be the concept for a specific instrument mapped to a LPS data tabulation model or the concept for a specific EHR system mapped to a clinical data tabulation model.

Use SURVEY_CONDUCT.response_visit_occurrence_id to refer to a clinic visit that is precipitated by an LPS visit.

A de novo vocabulary for longitudinal population studies

Introduction

Currently the de novo longitudinal population studies (LPS) vocabulary has two domains – demographics and protocol. Concepts in the demographic's domain fall into one of two classes: migration and other demographic constructs. The migration class covers several concepts associated with in-migration and out-migration. All of those concepts take categories.

As far as protocol goes, just like in clinical research, in longitudinal and cross-sectional observational research there is an order of events. This "order of events" refers to a protocol in which surveys are asked and labs are performed.

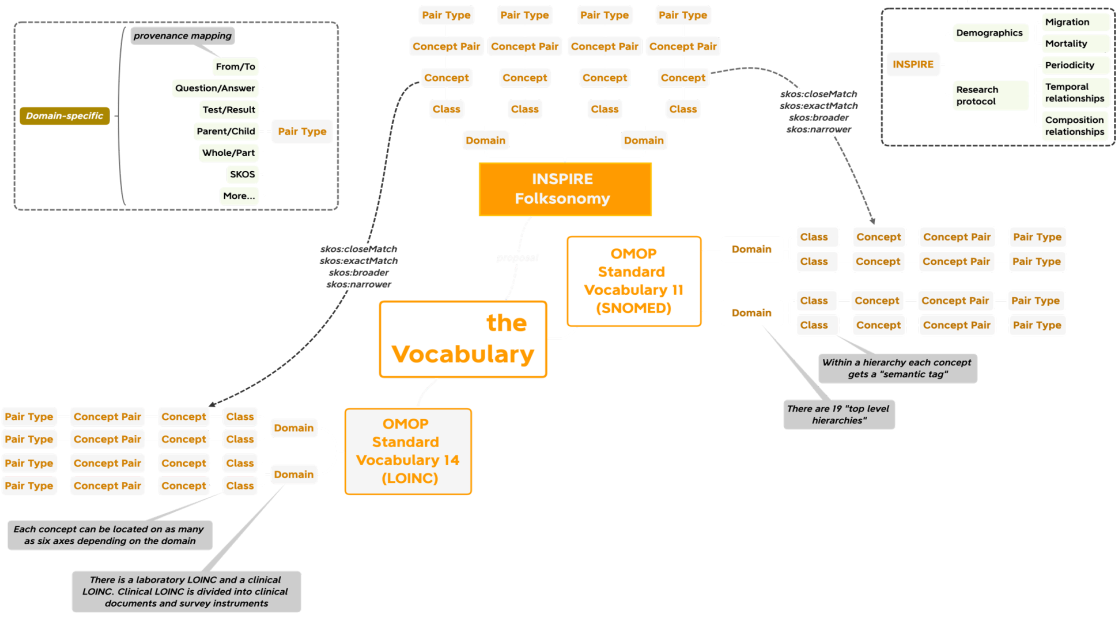
As it turns out, as part of the OMOP CDM, we will need a domain for describing the course of surveys and labs in the context of SURVEY_CONDUCTs, VISIT_OCCURRENCEs and VISIT_DETAILs. This is the protocol domain.

Concepts in the protocol domain fall into one of three classes: periodicity, temporal relationships and other protocol constructs. Periodicity refers to the frequency with which instruments and labs are asked from one round to the next. Periodicity takes categories. In SURVEY_CONDUCT there is a timing_concept_id that refers to a periodicity concept and its allowable categories.

In the past in the CDM Builders forum in connection with VISIT_OCCURRENCEs and VISIT_DETAILs there have been discussions of the need for temporal relationships generally and Allen's interval algebra specifically as a way of ordering occurrences. Temporal relationships are a second class into which protocol concepts fall in the LPS vocabulary. The third class is a catch all: other protocol constructs.

For now, it is imagined that this vocabulary will be more like a folksonomy than a taxonomy: the organization of CONCEPTs will be a little ad hoc and largely flat. In folksonomies, hierarchic taxonomies, however, sometimes follow.

Note that in addition to the construction of several "controlled vocabularies" (DDI) each of which includes a variable concept and one or more category concepts together with their CONCEPT_RELATIONSHIPs, implementing the LPS vocabulary entails creating one or more records in other OMOP vocabulary tables: 1 VOCABULARY record, 2 DOMAIN records and 6 CONCEPT_CLASS records. A complete accounting of "controlled vocabularies" follows.


provenance mapping
From/To
Question/Answer
Test/Result
Parent/Child
Whole/Part
SKOS
More...
Pair Type
Domain-specific
Pair Type
Concept Pair
Concept
Class
Domain
INSPIRE Folksonomy
skos:closeMatch
skos:exactMatch
skos:broader
skos:narrower
INSPIRE
Demographics
Migration
Mortality
Periodicity
Research protocol
Temporal relationships
Composition relationships
the Vocabulary
OMOP Standard Vocabulary 11 (SNOMED)
OMOP Standard Vocabulary 14 (LOINC)
Within a hierarchy each concept gets a "semantic tag"
There are 19 "top level hierarchies"
Each concept can be located on as many as six axes depending on the domain
There is a laboratory LOINC and a clinical LOINC. Clinical LOINC is divided into clinical documents and survey instruments

The demographics domain

Migration

Migration hosts two codelists aka category sets – one for the OMOP OBSERVATION that is mapped to entry_type from the ALPHA source spec and the other for the OBSERVATION that is mapped to exit_type. In each OBSERVATION OBSERVATION.observation_concept_id is a foreign key into either the entry_type concept or the exit_type concept in the CONCEPT table. Entry_type in the ALPHA spec is the start of a residency episode. Exit_type is the end of a residency episode. We are creating concepts for each of these events in the de novo LPS vocabulary as well as a FACT_RELATIONSHIP through which these events become the bookends of a residency episode. We are also creating concepts for the categories these events take. Across Health Demographic and Surveillance Systems in Africa and elsewhere a has evolved.

Other demographic constructs

The protocol domain

Periodicity

Currently just one concept falls under periodicity that is applicable to surveys. It is **survey periodicity**. The **survey periodicity** concept is referenced by SURVEY_CONDUCT.timing_concept_id. The **survey periodicity** concept takes several categories: **cross-sectional**, monthly **follow-up**, **quarterly follow-up**, **semi-annual follow-up**, **annual follow-up, quota-driven follow-up**, **special follow-up**. Each of these categories is a concept in the de novo vocabulary along with **survey periodicity**. In the CONCEPT_RELATIONSHIP table **survey periodicity** *subsumes* each of these categories.

Eventually it is anticipated that certain category sets specific to the periodicity of sensors will fall under periodicity.

Temporal relationships

Allen's interval algebra is optionally referenced by an OBSERVATION.observation_concept_id in the event we want to extend VISIT_DETAIL to describe the order of a person's visit details within a visit in a more nuanced than the standard visit detail chains support. **Allen's interval algebra** falls under temporal relationships. It *subsumes* 13 **allen relationship**s where each **allen relationship** is its own concept, and in the CONCEPT_RELATIONSHIP table there is a *subsumes* record that goes between **Allen's interval algebra** and each of the 13 **allen relationship**s.

Other protocol constructs

We will create a new VOCABULARY called **Data Tabulation Models** aka CRFs**.** All concepts in this vocabulary will have a domain of "Data Tabulation Models". Models will get one of three classifications for now: Clinical, Longitudinal Population Study and Patient Registry. There will be a CONCEPT in this VOCABULARY for each on-ramp specification. That includes the WHO

COVID-19 CORE CRF, the IHCC/WT COVID-19 LMIC Questionnaire, the WHO Verbal Autopsy and so forth. The SURVEY_CONDUCT **study_concept_id** references one of these concepts.

The **survey respondent type** concept is referenced by the SURVEY_CONDUCT.respondent_type_concept_id. The **survey respondent type** takes one of six answers: **self**, **family member**, **interviewer**, **neighbor**, **guardian**. **other respondent type**. In the CONCEPT_RELATIONSHIP table there is a record for each of the answers that **survey respondent type** takes.

The **visit type** concept is referenced by the VISIT_OCCURRENCE.visit_type_concept_id and the VISIT_DETAIL.visit_type_concept_id. In the context of an LPS visit type is either an **instrument**, an **instrument section** or a **lab**.

The **data tabulation model provenance** concept is referenced by OBSERVATION.observation_type_concept_id, **Data tabulation model provenance** take concepts for each of the several methodologies employed by HDSSs to convert source data into the actual format identified by the **data tabulation model**.

References

https://www.ohdsi.org/

https://ohdsi.github.io/TheBookOfOhdsi/

https://www.ohdsi.org/software-tools/

http://ohdsi.github.io/WhiteRabbit/WhiteRabbit.html#installation_and_support

http://ohdsi.github.io/WhiteRabbit/RabbitInAHat.html#installation_and_support

http://ohdsi.github.io/WhiteRabbit/WhiteRabbit.html

https://ohdsi.github.io/TheBookOfOhdsi/ExtractTransformLoad.html#rabbit-in-a-hat

http://ohdsi.github.io/WhiteRabbit/RabbitInAHat.html

https://github.com/OHDSI/WhiteRabbit