LTS Computing LLC

Common Data Model (CDM v5.3.1)

ETL Mapping Specification for OpenMRS

Theresa Yu & Lee Evans

Version 1.0

September 2, 2018

Table of Contents

[Introduction 2](#_Toc50475762)

[Source Data Mapping Approach 3](#_Toc50475763)

[Source to Standard Concept Mapping 4](#_Toc50475764)

[Miscellaneous Mapping 5](#_Toc50475765)

[Location 6](#_Toc50475766)

[Care\_Site 7](#_Toc50475767)

[Provider 8](#_Toc50475768)

[Person 9](#_Toc50475769)

[Observation Period 11](#_Toc50475770)

[Visit Occurrence 12](#_Toc50475771)

[Condition Occurrence 14](#_Toc50475772)

[Condition Type Concept id 16](#_Toc50475773)

[Procedure Occurrence 17](#_Toc50475774)

[Procedure Type Concept id 19](#_Toc50475775)

[Device Exposure 20](#_Toc50475776)

[Drug Exposure 21](#_Toc50475777)

[Measurement 24](#_Toc50475778)

[Observation 26](#_Toc50475779)

[Death 29](#_Toc50475780)

[Fact Relationship 30](#_Toc50475781)

# Introduction

In 2015, members of the OHDSI community created a proof of concept project to load OpenMRS demo data to the OHDSI Common Data Model. Mapping specifications for an ETL process was created. The versions that they worked with were OpenMRS v1.6 and CDM v 5.0. This prototype ETL specification is based on the original specifications but upgraded to be compatible with OpenMRS v1.9 and CDM v 5.3.1. It is also based on a very small sample simple synthetic test dataset that was provided by OpenMRS.

The approach taken for this prototype is to first export openMRS data csv files using mySQL client and load them into OHDSI OMOP CDM staging tables using postgresql client. The two database clients are made available in a Docker container image. Once the openMRS data is in the OHDSI OMOP CDM staging tables the ETL SQL scripts are run to convert and populate the data into the postgresql OHDSI OMOP CDM tables. The OHDSI Atlas observational web application can then be used to analyze the data in the ohdSI OMOP CDM tables.

**Software and Documentation Copyright & License**

Copyright 2020 LTS Computing LLC

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

# Source Data Mapping Approach

Person

Patient

Patient Identifier

Person Address

Location

Visit

Encounter

Encounter Provider

Provider

Obs

Active List

Active List Allergy

Orders

Relationship

Person

Location

Observation Period

Care Site

Visit Occurrence

Provider

Procedure Occurrence

Condition Occurrence

Drug Exposure

Device Exposure

Measurement

Observation

Death

Fact Relationship

# Source to Standard Concept Mapping

The query below is used to create standard mappings from OpenMRS CIEL codes to SNOMED and RxNorm which are considered standard concepts in CDM. The query uses the CDM Concept and Concept Relationship and Source To Concept Map tables. The results are stored in a temporary table and used to populate concept id in different event tables including Condition Occurrence, Procedure Occurrence, Drug Exposure etc.

SELECT DISTINCT cdm\_src.domain\_id as source\_domain\_id,

c1.domain\_id as target\_domain\_id,

cdm\_src.vocabulary\_id as source\_vocabulary\_id,

c1.VOCABULARY\_ID as target\_vocabulary\_id,

mrs\_src.concept\_id AS source\_concept\_code,

cdm\_src.concept\_id as source\_concept\_id,

CASE WHEN c1.concept\_id is null THEN 0 else c1.concept\_id END as target\_concept\_id,

cdm\_src.valid\_start\_date,

cdm\_src.valid\_end\_date,

cdm\_src.concept\_name,

c1.concept\_name

FROM stg.concept mrs\_src

LEFT JOIN cdm.concept cdm\_src ON mrs\_src.concept\_id = cast(cdm\_src.concept\_code as integer) AND cdm\_src.vocabulary\_id = 'CIEL'

LEFT JOIN cdm.concept\_relationship cr ON cdm\_src.concept\_id = cr.concept\_id\_1 AND cr.invalid\_reason IS NULL AND cr.relationship\_id = 'Maps to'

LEFT JOIN cdm.concept c1 ON cr.concept\_id\_2 = c1.concept\_id AND c1.invalid\_reason is null and c1.standard\_concept IS NOT NULL

WHERE mrs\_src.concept\_id not in (select m.source\_code::integer from source\_to\_concept\_map m where m.source\_vocabulary\_id in ('OpenMRS\_Drug','OpenMRS\_Condition') )

UNION

SELECT c1.domain\_id as source\_domain\_id,

c2.domain\_id as target\_domain\_id,

scm.source\_vocabulary\_id as source\_vocabulary\_id,

c2.vocabulary\_id as target\_vocabulary\_id,

scm.source\_code::integer AS source\_concept\_code,

scm.source\_concept\_id as source\_concept\_id,

scm.target\_concept\_id,

scm.valid\_start\_date,

scm.valid\_end\_date,

source\_code\_description,

c2.concept\_name

FROM cdm.source\_to\_concept\_map scm

JOIN cdm.concept c1 on c1.concept\_id = scm.source\_concept\_id

JOIN cdm.concept c2 on c2.concept\_id = scm.target\_concept\_id

WHERE scm.source\_vocabulary\_id in ('OpenMRS\_Drug','OpenMRS\_Condition');

# Miscellaneous Mapping

The query below is used to create mapping from CIEL units of measure (e.g. ft, kg, weeks) and operators (e.g. =, <=, >). The query uses the CDM Concept and Source To Concept Map tables. The results are stored in a temporary table and used to populate unit and modifier concept ids in the CDM Measurement and Observation table.

SELECT concept\_name,

concept\_id,

concept\_code,

valid\_start\_date,

valid\_end\_date,

invalid\_reason

FROM cdm.concept

WHERE (vocabulary\_id = 'UCUM' or (vocabulary\_id = 'SNOMED' and domain\_id = 'Meas Value Operator')) and standard\_concept = 'S';

UNION

SELECT source\_code\_description as concept\_name,

target\_concept\_id as concept\_id,

source\_code as concept\_code,

valid\_start\_date,

valid\_end\_date,

invalid\_reason

FROM cdm.source\_to\_concept\_map

WHERE source\_vocabulary\_id='OpenMRS\_UOM'

AND source\_code\_description not in (select concept\_name from uom\_concept\_mapping\_tmp001);

# Location

Populate location table from OpenMRS location table. Include only records where at least 1 of the address fields – name, address1, address2, city\_village, state\_province, postal\_code - is populated.

(This shows how address data could be mapped but the recommendation is that, to maintain patient anonymity, do not store the actual names and address fields from OpenMRS person\_address table in the OMOP CDM location table – leave the name and address fields set to NULL)

|  |  |  |  |
| --- | --- | --- | --- |
| Destination Field | Source Field | Logic | Comment |
| location\_id |  | System generated |  |
| address\_1 | name or address1 | If address 1 is blank, use name, else use address1 |  |
| address\_2 | address2 |  |  |
| city | city\_village |  |  |
| state | state\_province |  |  |
| zip | postal\_code |  |  |
| county |  | Null |  |
| location\_source\_value |  | ‘L ’ || location\_id | Concatenate ‘L ‘ and location\_id |

Populate location table from OpenMRS person\_address table. Take the most recent person address by getting the max of person\_address\_id for each person.

| Destination Field | Source Field | Logic | Comment |
| --- | --- | --- | --- |
| location\_id |  | System generated |  |
| address\_1 | address1 |  | Get address corresponding to max(person\_address\_id) of each person |
| address\_2 | address2 |  |  |
| city | city\_village |  |  |
| state | state\_province |  |  |
| zip | postal\_code |  |  |
| county |  | Null |  |
| location\_source\_value |  | ‘P ’ || person\_address\_id | Concatenate ‘P ‘ and person\_address\_id |

# Care\_Site

Populate care\_site from OpenMRS location table.

Create a dummy care\_site record.

|  |  |  |  |
| --- | --- | --- | --- |
| Destination Field | Source Field | Logic | Comment |
| care\_site\_id |  | 0 |  |
| care\_site\_name |  | Null |  |
| place\_of\_service\_concept\_id |  | Null |  |
| location\_id |  | 0 |  |
| care\_site\_source\_value |  | Null |  |
| place\_of\_service\_source\_value |  | Null |  |

Create a care site record for locations used as care sites. The assumption is if the location exists in the OpenMRS encounter table, the location is a care site.

|  |  |  |  |
| --- | --- | --- | --- |
| Destination Field | Source Field | Logic | Comment |
| care\_site\_id | Location.  Locationid | Check OpenMRS encounter table to check locationid is used in an encounter. | Use locationid from OpenMRS location table. |
| care\_site\_name | Location.name |  |  |
| place\_of\_service\_concept\_id | 0 |  |  |
| location\_id | Location.locationid | Join OpenMRS location table to CDM location table on OpenMRS locationid = substring (location\_source\_value,3) | Use Location Id from CDM location table  CDM location table stores the OpenMRS locationid in the location\_source\_value. It is concatenated to ‘L ‘ which designates the source table (OpenMRS location) so the actual locationid starts in position 3. |
| care\_site\_source\_value | Location.locationid | Null | Use locationid from OpenMRS location table |
| place\_of\_service\_source\_value |  | Null |  |

# Provider

Populate provider table from OpenMRS provider table. Since providers are also stored in the person table, join to OpenMRS person table to get the provider’s gender and of year of birth.

|  |  |  |  |
| --- | --- | --- | --- |
| Destination Field | Source Field | Logic | Comment |
| provider\_id | provider.provider\_id |  |  |
| provider\_name | provider.name |  |  |
| Npi |  | Null |  |
| Dea |  | Null |  |
| specialty\_concept\_id |  | Null |  |
| care\_site\_id |  | Null |  |
| year\_of\_birth | year(person.  birthdate) | Join OpenMRS provider table to OpenMRS person table on provider\_id = person\_id to get birthdate. |  |
| gender\_concept\_id |  | Join OpenMRS provider table to OpenMRS person table on provider\_id = person\_id to get gender. Based on value of gender, assign the following concept\_id:  F = 8532  M= 8507  Else 0 |  |
| provider\_source\_value | provider.provider\_id | 38004514 | Unknown Physician Specialty |
| specialty\_source\_value |  | Null |  |
| specialty\_source\_concept\_id |  | Null |  |
| gender\_source\_value | person.gender |  |  |
| gender\_source\_concept\_id |  | Null |  |

# Person

Populate person from OpenMRS person table. Include only the records where voided = 0 and birthdate >= 1900.

| Destination Field | Source Field | Logic | Comment |
| --- | --- | --- | --- |
| person\_id | Person.person\_id |  |  |
| gender\_concept\_id | Person.gender | Based on value of gender, assign the following:  F = 8532  M= 8507  Else 0 |  |
| year\_of\_birth | birthdate | YEAR(birthdate) |  |
| month\_of\_birth | birthdate | MONTH(birthdate) |  |
| day\_of\_birth | birthdate | DAY(birthdate) |  |
| birth\_datetime | birthdate |  |  |
| race\_concept\_id |  | 0 | Race is an attribute that can be stored in the person\_attribute\_  type table. We are not mapping attributes in this release |
| ethnicity\_concept\_id |  | 0 | Ethnicity is an attribute that can be stored in the person\_attribute\_  type table. We are not mapping attributes in this release |
| location\_id | Location.location\_id | Join to person\_address table on person\_id to get person\_address\_id. Find locationid from CDM location table. Join on person\_address\_id = substring(location\_source  \_value ,3) | CDM location table stores the OpenMRS person\_address\_id in the location\_source\_value. It is concatenated to  ‘P ‘ which designates the source table OpenMRS person\_address) so the actual locationid starts in position 3. |
| provider\_id |  | Null |  |
| care\_site\_id |  | Null |  |
| person\_source\_value | person.person\_id || ‘ ‘ || patient\_identifier.identifier | Join MRS person table to patient\_identifier table on person\_id = patient\_id. Concatenate person.person\_id and patient\_identifier.identifier. | Add a space between person\_id and identifier |
| gender\_source\_value | person.gender |  |  |
| gender\_source\_concept\_id |  | Null |  |
| race\_source\_value |  | Null |  |
| race\_source\_concept\_id |  | Null |  |
| ethnicity\_source\_value |  | Null |  |
| ethnicity\_source\_concept\_id |  | Null |  |

# Observation Period

Populate observation\_period from OpenMRS obs, orders, and active\_list tables. Use start and end dates from each table and take the earliest and latest date values for each person.

|  |  |  |  |
| --- | --- | --- | --- |
| Destination Field | Source Field | Logic | Comment |
| observation\_period\_id |  |  |  |
| person\_id | person\_id |  |  |
| observation\_period\_start\_date | obs.obs\_datetime  or  orders.date\_activated  or  active\_list.start\_date | For each person, find the min\_value of either the obs.obs\_datetime, orders.date\_activated, or active\_list.start\_date |  |
| observation\_period\_end\_date | obs.obs\_datetime  or  orders.date\_stopped  or  active\_list.end\_date | For each person, find the max\_value of either the obs.obs\_datetime, orders.date\_stopped, or active\_list.end\_date. If orders\_date\_stopped is not populated, use orders.auto\_expire\_date |  |
| period\_type\_concept\_id |  | 44814725 | Period inferred by algorithm |

# Visit Occurrence

Populate visit\_occurrence from OpenMRS visit and encounter tables. In OpenMRS encounters can be within a visit or an encounter can stand alone with no matching visit. Since CDM only stores 1 visit occurrence record for multiple visit/encounter records covering the same date range, only the first provider id and encounter type is stored. For the sample data used for this release, there are no visit records and all encounters are of type outpatient. No collapsing logic was implemented.

(Future release will include collapsing contiguous inpatient or ER visits into one inpatient visit. It will also include using the visit\_detail table which will allow encounter record details like provider and type to be stored.)

Below are the mappings for OpenMRS visit table.

| Destination Field | Source Field | Logic | Comment |
| --- | --- | --- | --- |
| visit\_occurrence\_id |  | System generated |  |
| person\_id | patient\_id |  |  |
| visit\_concept\_id | source\_to\_concept\_map.  target\_concept\_id | Join visit table to source\_to\_concept\_map on visit\_type = source\_code and source vocabulary id = ‘OpenMRS\_Visit\_Type’. | Set to 0 if no mapping found |
| visit\_start\_date | date\_started | Extract date part | Ensure date\_started is within person’s observation period |
| visit\_start\_datetime | date\_started |  |  |
| visit\_end\_date | date\_stopped | Extract date part | Ensure date\_stopped is within person’s observation period |
| visit\_end\_datetime | date\_stopped |  |  |
| visit\_type\_concept\_id |  | 44818518 | 44818518 Visit derived from EHR record |
| provider\_id |  | Null | OpenMRS visit does not have provider id |
| care\_site\_id | location\_id |  |  |
| visit\_source\_value | visit\_type\_id |  |  |
| visit\_source\_concept\_id |  | Null |  |
| admitting\_source\_concept\_id |  | Null |  |
| admitting\_source\_value |  | Null |  |
| discharge\_to\_concept\_id |  | Null |  |
| discharge\_to\_concept\_id |  | Null |  |
| preceding\_visit\_occurrence\_id |  | Null |  |

Below are the mappings for OpenMRS encounter table.

| Destination Field | Source Field | Logic | Comment |
| --- | --- | --- | --- |
| visit\_occurrence\_id |  | System generated |  |
| person\_id | patient\_id |  |  |
| visit\_concept\_id | source\_to\_concept\_map.  target\_concept\_id | Join visit table to source\_to\_concept\_map on visit\_type = source\_code using source vocabulary id OpenMRS\_EncounterTyp | Set to 0 if no mapping found |
| visit\_start\_date | encounter\_datetime | Extract date part | Ensure encounter\_datetime is within person’s observation period |
| visit\_start\_datetime | encounter\_datetime |  |  |
| visit\_end\_date | encounter\_datetime | Extract date part |  |
| visit\_end\_datetime | encounter\_datetime |  |  |
| visit\_type\_concept\_id |  | 44818518 | 44818518 Visit derived from EHR record |
| provider\_id | provider\_id |  | For encounters with the same encounter\_datetime and visit\_concept\_id, get the first provider\_id |
| care\_site\_id | location\_id |  |  |
| visit\_source\_value | encounter\_type |  | For encounters with the same encounter\_datetime and visit\_concept\_id, get the first encounter type |
| visit\_source\_concept\_id |  | Null |  |
| admitting\_source\_concept\_id |  | Null |  |
| admitting\_source\_value |  | Null |  |
| discharge\_to\_concept\_id |  | Null |  |
| discharge\_to\_concept\_id |  | Null |  |
| preceding\_visit\_occurrence\_id |  | Null |  |

# Condition Occurrence

Populate condition\_occurrence from OpenMRS obs and active\_list table.

Below are the mappings for OpenMRS obs table.

| Destination Field | Source Field | Logic | Comment |
| --- | --- | --- | --- |
| condition\_occurrence\_id |  | System generated |  |
| person\_id | obs.person\_id |  |  |
| condition\_concept\_id | target\_concept\_id | Use the source\_to\_standard\_  concept query.  Join obs.value\_coded to source\_concept\_  code. Assign target\_concept\_id to condition\_concept\_id. | Include only where the target domain is ‘Condition’. If no standard mapping is found (target\_concept\_id is null), include if the source\_domain\_id is ‘Condition’. |
| condition\_start\_date | obs\_datetime | extract date from obs\_datetime |  |
| condition\_start\_datetime | obs\_datetime |  |  |
| condition\_end\_date |  | Null |  |
| condition\_end\_datetime |  | Null |  |
| condition\_type\_concept\_id |  | Use the source\_to\_standard\_  concept query.  Join obs.concept\_id to source\_concept\_  code. Depending on source\_concept\_id, assign condition\_type\_concept\_id. | See condition type concept id table below. |
| stop\_reason |  | Null |  |
| provider\_id | encounter\_provider.  provider\_id | Join to encounter.provider using encounter\_id. |  |
| visit\_occurrence\_id | visit\_occurrence.  visit\_occurrence\_id | Use obs\_datetime and encounter\_type to lookup visit\_occurrence\_id in CDM visit\_occurrence table. | obs,datetime must be within visit start and end date.  Look up encounter\_tupe in table source\_to\_concept\_map and use the target\_concept\_id to match the visit\_concept\_id |
| condition\_source\_value | value\_coded |  |  |
| condition\_source\_concept\_id | source\_concept\_id |  | From source\_to\_standard\_  concept query. |
| condition\_status\_source\_value |  | Null |  |
| condition\_status\_concept\_id |  | Null |  |

Below are the mappings for OpenMRS active\_list table.

|  |  |  |  |
| --- | --- | --- | --- |
| Destination Field | Source Field | Logic | Comment |
| condition\_occurrence\_id |  | System generated |  |
| person\_id | active\_list.person\_id |  |  |
| condition\_concept\_id | target\_concept\_id | Use the source\_to\_standard\_  concept query.  Join active\_list.concept\_id to source\_concept\_  code. Assign target\_concept\_id to condition\_concept\_id. | Include only where the target domain is ‘Condition’. If no standard mapping is found (target\_concept\_id is null), include if the source\_domain\_id is ‘Condition’. |
| condition\_start\_date | start\_date | extract date from obs\_datetime |  |
| condition\_start\_datetime | start\_date |  |  |
| condition\_end\_date | end\_date | Null |  |
| condition\_end\_datetime | end\_date | Null |  |
| condition\_type\_concept\_id |  | 43542353 | Observation recorded from EHR |
| stop\_reason |  | Null |  |
| provider\_id |  | Null |  |
| visit\_occurrence\_id |  | Null |  |
| condition\_source\_value | active\_list.concept\_id |  |  |
| condition\_source\_concept\_id | source\_concept\_id |  | From source\_to\_standard\_  concept query. |
| condition\_status\_source\_value |  | Null |  |
| condition\_status\_concept\_id |  | Null |  |

## Condition Type Concept id

|  |  |  |  |
| --- | --- | --- | --- |
| Target Concept Id | Target Concept Name | Source Concept Id | Source Concept Name |
| 38000245 | EHR problem list entry | 45947502 | Reason antiretrovirals stopped |
| 45947507 | Reason tuberculosis treatment stopped |
| 45933647 | Reason tuberculosis prophylaxis stopped |
| 45933646 | Reason PCP prophylaxis stopped |
| 45922681 | CC (Chief Complaint) | 42894222 | EHR chief complaint |
| 43542353 | Observation recorded from EHR | 45906319 | Childs current HIV status |
| 45906183 | General exam findings |
| 45919934 | Peds CDC specific condition query |
| 45947498 | Peds WHO category query |
| 45912169 | Peds CDC category query |
| 45922789 | Adult who condition query |
| 45912294 | Current who HIV stage |
| 5086 | Condition tested for by diagnostic procedure | 45952023 | Review of systems, general |
| 45906178 | Review of systems, cardiopulmonary |
| 45947476 | Review of systems, gastrointestinal |
| 45952026 | Review of systems, genitourinary |
| 45947477 | Review of systems, musculoskeletal |
| 45947473 | Review of systems, HEENT |
| 32020 | EHR encounter diagnosis | 45922787 | Diagnosis added |
| 45952031 | HEENT exam findings |
| 45912162 | Mucocutaneous exam findings |
| 45922407 | X-ray, chest |
| 43542353 | Observation recorded from EHR (Condition) | All other not listed above |  |

# Procedure Occurrence

Populate condition\_occurrence from OpenMRS obs and orders table.

Below are the mappings for OpenMRS obs table.

| Destination Field | Source Field | Logic | Comment |
| --- | --- | --- | --- |
| procedure\_occurrence\_id |  | System generated |  |
| person\_id | obs.person\_id |  |  |
| procedure\_concept\_id | target\_concept\_id | Use the source\_to\_standard\_  concept query.  Join obs.value\_coded to source\_concept\_  code. Assign target\_concept\_id to procedure\_concept\_id. | Include only where the target domain is ‘Procedure’. If no standard mapping is found (target\_concept\_id is null), include if the source\_domain\_id is ‘Procedure’. |
| procedure\_date | obs\_datetime | extract date from obs\_datetime |  |
| procedure\_datetime | obs\_datetime |  |  |
| procedure\_type\_concept\_id |  | Use the source\_to\_standard\_  concept query.  Join obs.concept\_id to source\_concept\_  code. Depending on source\_concept\_id, assign procedure \_type\_concept\_i. | See procedure type concept id table below. |
| modifier\_concept\_id |  | Null |  |
| quantity |  | Null |  |
| provider\_id | encounter\_provider.  provider\_id | Join to encounter.provider using encounter\_id. |  |
| visit\_occurrence\_id | visit\_occurrence.  visit\_occurrence\_id | Use obs\_datetime and encounter\_type to lookup visit\_occurrence\_id in CDM visit\_occurrence table. | obs,datetime must be within visit start and end date.  Look up encounter\_tupe in table source\_to\_concept\_map and use the target\_concept\_id to match the visit\_concept\_id |
| procedure\_source\_value | value\_coded |  |  |
| procedure\_source\_concept\_id | source\_concept\_id |  | From source\_to\_standard\_  concept query. |
| qualifier\_source\_value |  | Null |  |

Below are the mappings for OpenMRS orders table.

|  |  |  |  |
| --- | --- | --- | --- |
| Destination Field | Source Field | Logic | Comment |
| procedure\_occurrence\_id |  | System generated |  |
| person\_id | orders.patient\_id |  |  |
| procedure\_concept\_id | target\_concept\_id | Use the source\_to\_standard\_  concept query.  Join orders.concept\_id to source\_concept\_  code. Assign target\_concept\_id to procedure\_concept\_id. | Include only where the target domain is ‘Procedure’. If no standard mapping is found (target\_concept\_id is null), include if the source\_domain\_id is ‘Procedure’. |
| procedure\_date | date\_activated | extract date from date\_activated |  |
| procedure\_datetime | date\_activated |  |  |
| procedure\_type\_concept\_id |  | 38000275 | EHR order |
| modifier\_concept\_id |  | Null |  |
| quantity |  | Null |  |
| provider\_id | encounter\_provider.  provider\_id | Join to encounter.provider using encounter\_id. |  |
| visit\_occurrence\_id | visit\_occurrence.  visit\_occurrence\_id | Use obs\_datetime and encounter\_type to lookup visit\_occurrence\_id in CDM visit\_occurrence table. | obs,datetime must be within visit start and end date.  Look up encounter\_tupe in table source\_to\_concept\_map and use the target\_concept\_id to match the visit\_concept\_id |
| procedure\_source\_value | orders.concept\_id | Null |  |
| procedure\_source\_concept\_id | source\_concept\_id |  | From source\_to\_standard\_  concept query. |
| qualifier\_source\_value |  | Null |  |

## Procedure Type Concept id

| Target Concept Id | Target Concept Name | Source Concept Id | Source Concept Name |
| --- | --- | --- | --- |
| 42898141 | Referral record | 45933648 | REFERRALS ORDERED |
| 38000275 | EHR order list entry | 45941938 | CRYPTOCOCCAL TREATMENT PLAN |
| 45947503 | ANTIRETROVIRAL PLAN |
| 45952047 | TUBERCULOSIS PROPHYLAXIS PLAN |
| 45906199 | TUBERCULOSIS TREATMENT PLAN |
| 45947508 | TESTS ORDERED |
| 45922555 | ANTIRETROVIRAL USE DURING PREGNANCY |
| 45922569 | REASON ANTIRETROVIRALS STARTED |
| 45952033 | OVERALL DRUG ADHERENCE IN LAST MONTH |
| 45947494 | PREVIOUS IMMUNIZATIONS ADMINISTERED |
| 45935822 | IMMUNIZATIONS |
| 45947505 | PCP PROPHYLAXIS PLAN |
| 38000275 | EHR order list entry | All other not listed above |  |

# Device Exposure

Populate device\_exposure from OpenMRS obs table.

|  |  |  |  |
| --- | --- | --- | --- |
| Destination Field | Source Field | Logic | Comment |
| device\_exposure\_id |  | System generated |  |
| person\_id | person\_id |  |  |
| device\_concept\_id | target\_concept\_id | Use the source\_to\_standard\_  concept query.  Join obs.value\_coded to source\_concept\_  code. Assign target\_concept\_id to device\_concept\_id. | Include only where the target domain is ‘Device’. If no standard mapping is found (target\_concept\_id is null), include if the source\_domain\_id is ‘Device’. |
| device\_exposure\_start\_date | obs\_datetime | extract date from OBS\_DATETIME |  |
| device\_exposure\_start  datetime | obs\_datetime |  |  |
| device\_exposure\_end\_date |  | Null |  |
| device\_exposure\_end\_  datetime |  | Null |  |
| device\_type\_concept\_id |  | 44818707 | EHR Detail |
| unique\_device\_id |  | Null |  |
| quantity |  | Null |  |
| provider\_id | encounter\_provider.  provider\_id | Join to encounter.provider using encounter\_id. |  |
| visit\_occurrence\_id | visit\_occurrence.  visit\_occurrence\_id | Use obs\_datetime and encounter\_type to lookup visit\_occurrence\_id in CDM visit\_occurrence table. | obs,datetime must be within visit start and end date.  Look up encounter\_tupe in table source\_to\_concept\_map and use the target\_concept\_id to match the visit\_concept\_id |
| device\_source\_value | value\_coded |  |  |
| device\_source\_concept\_id | source\_concept\_id |  | From source\_to\_standard\_  concept query. |

# Drug Exposure

Populate drug\_exposure from OpenMRS obs and orders table.

(Future enhancement: join obs to orders table to get drug detail so all drug\_exposure records will come from obs table)

Below are the mappings for OpenMRS obs table.

| Destination Field | Source Field | Logic | Comment |
| --- | --- | --- | --- |
| drug\_exposure\_id |  | System generated |  |
| person\_id | person\_id |  |  |
| drug\_concept\_id | target\_concept\_id | Use the source\_to\_standard\_  concept query.  Join obs.value\_coded to source\_concept\_  code. Assign target\_concept\_id to drug\_concept\_id. | Include only where the target domain is ‘Drug’. If no standard mapping is found (target\_concept\_id is null), include if the source\_domain\_id is ‘Drug’. |
| drug\_exposure\_start\_date | obs\_datetime | extract date from obs.datetime |  |
| drug\_exposure\_  start\_datetime | obs\_datetime |  |  |
| drug\_exposure\_end\_date | obs\_datetime | extract date from obs.datetime |  |
| drug\_exposure\_  end\_datetime | obs\_datetime |  |  |
| verbatim\_end\_date |  | Null |  |
| drug\_type\_concept\_id |  | Use the source\_to\_standard\_  concept query.  Join obs.concept\_id to source\_concept\_  code. Depending on source\_concept\_id, assign procedure \_type\_concept\_i. | See procedure type concept id table below. |
| stop\_reason |  | Null |  |
| refills |  | Null |  |
| quantity |  | Null |  |
| days\_supply |  | Null |  |
| sig |  | Null |  |
| route\_concept\_id |  | Null |  |
| lot\_number |  | Null |  |
| provider\_id | encounter\_provider.  provider\_id | Join to encounter.provider using encounter\_id. |  |
| visit\_occurrence\_id | visit\_occurrence.  visit\_occurrence\_id | Use obs\_datetime and encounter\_type to lookup visit\_occurrence\_id in CDM visit\_occurrence table. | obs,datetime must be within visit start and end date.  Look up encounter\_tupe in table source\_to\_concept\_map and use the target\_concept\_id to match the visit\_concept\_id |
| drug\_source\_value | value\_coded |  |  |
| drug\_source\_concept\_id | source\_concept\_id |  | From source\_to\_standard\_  concept query. |
| route\_source\_value |  | Null |  |
| dose\_unit\_source\_value |  | Null |  |

Below are the mappings for OpenMRS orders table.

| Destination Field | Source Field | Logic | Comment |
| --- | --- | --- | --- |
| drug\_exposure\_id |  | System generated |  |
| person\_id | patient\_id |  |  |
| drug\_concept\_id | target\_concept\_id | Use the source\_to\_standard\_  concept query.  Join orders.concept\_id to source\_concept\_  code. Assign target\_concept\_id to drug\_concept\_id. | Include only where the target domain is ‘Drug’. If no standard mapping is found (target\_concept\_id is null), include if the source\_domain\_id is ‘Drug’. |
| drug\_exposure\_start\_date | date\_activated | extract date from date\_activated |  |
| drug\_exposure\_  start\_datetime | date\_activated |  |  |
| drug\_exposure\_end\_date | date\_stopped | extract date from date\_stopped | If date\_stopped is not available, use auto\_expire\_date |
| drug\_exposure\_  end\_datetime | date\_stopped |  | If date\_stopped is not available, use auto\_expire\_date |
| verbatim\_end\_date |  | Null |  |
| drug\_type\_concept\_id |  | 581452 | Dispensed in Outpatient office |
| stop\_reason |  | Null |  |
| refills | drug\_order.num\_refills | Join to drug\_order table using order\_id. |  |
| quantity | drug\_order.quantity | Join to drug\_order table using order\_id. |  |
| days\_supply |  | Join to drug\_order to order\_frequency on drug\_order.frequency = order\_frequencey.order\_  frequency\_id.  If frequence\_per\_day > 0, calculate days supply as  drug\_order.quantity / frequency\_per\_day. Otherwise, assign null. |  |
| sig |  | Null |  |
| route\_concept\_id | concept.concept\_id | Use the source\_to\_standard\_  concept query.  Join orders.route\_  concept\_id to source\_concept\_  code. Assign target\_concept\_id to route\_concept\_id. |  |
| lot\_number |  | Null |  |
| provider\_id | encounter\_provider.  provider\_id | Join to encounter.provider using encounter\_id. |  |
| visit\_occurrence\_id | visit\_occurrence.  visit\_occurrence\_id | Use obs\_datetime and encounter\_type to lookup visit\_occurrence\_id in CDM visit\_occurrence table. | orders.date\_activated and date\_stopped must be within visit start and end date.  Look up encounter\_type in table source\_to\_  concept\_map and use the target\_concept\_id to match the visit\_concept\_id |
| drug\_source\_value | orders.concept\_id |  |  |
| drug\_source\_concept\_id | source\_concept\_id |  | From source\_to\_standard\_  concept query. |
| route\_source\_value | drug\_order.route |  |  |
| dose\_unit\_source\_value | drug\_order.dose\_units |  |  |

# Measurement

Populate measurement from OpenMRS obs table.

| Destination Field | Source Field | Logic | Comment |
| --- | --- | --- | --- |
| measurement\_id |  | System generated |  |
| person\_id | person\_id |  |  |
| measurement\_concept\_id | target\_concept\_id  (using concept\_id) | Use the source\_to\_  standard  concept query.  Join obs.concept\_id to source\_concept\_  code. Assign target\_concept\_id to measurement \_concept\_id. | Include only where the target domain is ‘Measurement’. If no standard mapping is found (target\_concept\_id is null), include if the source\_domain\_id is ‘Measurement’. |
| measurement\_date | obs\_datetime | extract date from obs\_datetime |  |
| measurement\_datetime | obs\_datetime |  |  |
| measurement\_type\_  concept\_id |  | 44818701 | From physical examination |
| operator\_concept\_id | concept\_id | Use the miscellaneous mapping query. Join value\_modifier to concept\_name and vocabulary\_id = ‘SNOMED’. Assign concept\_id to operator\_concept\_  id |  |
| value\_as\_number | value\_numeric |  |  |
| value\_as\_concept\_id | target\_concept\_id  (using value\_coded) | Use the source\_to\_  standard  concept query.  Join obs.value\_coded to source\_concept\_  code. Assign target\_concept\_id to value\_as\_concept  \_id. |  |
| unit\_concept\_id | concept\_id | Join to concept\_numeric to get units. Use the miscellaneous mapping query.Join units to concept\_code and vocabulary\_id = ‘UCUM’. Assign concept\_id to unit\_concept\_id |  |
| range\_low | concept\_numeric.  low\_normal | Join to concept\_numeric on concept\_id. |  |
| range\_high | concept\_numeric.  hi\_normal | Join to concept\_numeric on concept\_id. |  |
| provider\_id | encounter\_provider.  provider\_id | Join to encounter.provider using encounter\_id. |  |
| visit\_occurrence\_id | visit\_occurrence.  visit\_occurrence\_id | Use obs\_datetime and encounter\_type to lookup visit\_occurrence\_id in CDM visit\_occurrence table. | orders.date\_activated and date\_stopped must be within visit start and end date.  Look up encounter\_type in table source\_to\_  concept\_map and use the target\_concept\_id to match the visit\_concept\_id |
| measurement\_source\_value | obs.concept\_id |  |  |
| measurement\_source\_  concept\_id | source\_concept\_id |  | From source\_to\_standard\_  concept query using obs.concept\_id |
| unit\_source\_value | concept\_numeric.  units | Join to concept\_numeric on concept\_id. |  |
| value\_source\_value | value\_numeric |  |  |

# Observation

Populate observation from OpenMRS obs, active\_list, and active\_list\_allergy tables.

Below are the mappings for OpenMRS obs table.Generated

| Destination Field | Source Field | Logic | Comment |
| --- | --- | --- | --- |
| observation\_id |  | System generated |  |
| person\_id | person\_id |  |  |
| observation\_concept\_id | target\_concept\_id  (using concept\_id) | Use the source\_to\_  standard  concept query.  Join obs.concept\_id to source\_concept\_  code. Assign target\_concept\_id to drug\_concept\_id. | Include only where the target domain is ‘Observation’. If no standard mapping is found (target\_concept\_id is null), include if the source\_domain\_id is ‘Observation’. |
| observation\_date | obs\_datetime | extract date from obs\_datetime |  |
| observation\_datetime | obs\_datetime |  |  |
| observation\_type\_concept\_id |  | 38000280 | Observation recorded from EHR |
| value\_as\_number | value\_numeric |  |  |
| value\_as\_string | value\_text  or  value\_datetime  or  value\_boolean  or value\_drug | Use the first non null value of either value\_text, value\_datetime, value\_boolean or value\_drug |  |
| value\_as\_concept\_id | target\_concept\_id  (using value\_coded) | Use the source\_to\_  standard  concept query.  Join obs.value\_coded to source\_concept\_  code. Assign target\_concept\_id to value\_as\_concept  \_id. |  |
| qualifier\_concept\_id |  | Null |  |
| unit\_concept\_id |  | Use the miscellaneous mapping query. Join value\_modifier to concept\_name and vocabulary\_id = ‘SNOMED’. Assign concept\_id to unit\_concept\_id |  |
| provider\_id | encounter\_provider.  provider\_id | Join to encounter.provider using encounter\_id. |  |
| visit\_occurrence\_id | visit\_occurrence.  visit\_occurrence\_id | Use obs\_datetime and encounter\_type to lookup visit\_occurrence\_id in CDM visit\_occurrence table. | obs.datetime must be within visit start and end date.  Look up encounter\_type in table source\_to\_  concept\_map and use the target\_concept\_id to match the visit\_concept\_id |
| observation\_source\_value | obs.concept\_id |  |  |
| observation\_source\_concept\_id | source\_concept\_id |  | From source\_to\_standard\_  concept query using obs.concept\_id |
| unit\_source\_value | concept\_numeric.  units | Join to concept\_numeric on concept\_id. |  |
| qualifier\_source\_value |  | Null |  |

Below are the mappings for OpenMRS active\_list and active\_list\_allergy tables. To get active\_list\_allergy records, Join active\_list to active\_list\_allergy on active\_list\_id.Generated

| Destination Field | Source Field | Logic | Comment |
| --- | --- | --- | --- |
| observation\_id |  | System generated |  |
| person\_id | person\_id |  |  |
| observation\_concept\_id | target\_concept\_id | Use the source\_to\_standard\_  concept query.  **active\_list:**  Join active\_list.concept\_id to source\_concept\_  code.  **active\_list\_allergy:**  Join active\_list\_allergy  .reaction\_concept\_id to source\_concept\_  code.  Assign target\_concept\_id to observation\_concept  \_id. | Include only where the target domain is ‘Observation’. If no standard mapping is found (target\_concept\_id is null), include if the source\_domain\_id is ‘Observation’. |
| observation\_date | start\_date | extract date from  start \_date |  |
| observation\_datetime | start\_date |  |  |
| observation\_type\_concept\_id |  | 38000280 | Observation recorded from EHR |
| value\_as\_number |  | Null |  |
| value\_as\_string |  | Null |  |
| value\_as\_concept\_id |  | Null |  |
| qualifier\_concept\_id |  | Null |  |
| unit\_concept\_id |  | Null |  |
| provider\_id |  | Null |  |
| visit\_occurrence\_id |  | Null |  |
| observation\_source\_value | **active\_list:**  concept\_id  **active\_list\_allergy:**  reaction\_concept\_id |  |  |
| observation\_source\_concept\_id | source\_concept\_id |  | From source\_to\_standard\_  concept query. |
| unit\_source\_value |  | Null |  |
| qualifier\_source\_value |  | Null |  |

# Death

Populate death from OpenMRS person table where death date is not null.

|  |  |  |  |
| --- | --- | --- | --- |
| Destination Field | Source Field | Logic | Comment |
| person\_id | person\_id |  |  |
| death\_date | death\_date |  | Not null |
| death\_type\_concept\_id | 38003566 |  | EHR record patient status "Deceased" |
| cause\_concept\_id | target\_concept\_id | Use the source\_to\_  standard  concept query.  Join cause\_concept\_id  to source\_concept\_  code. Assign target\_concept\_id to cause\_concept\_id. | If target\_concept\_id is null, set to 0. |
| cause\_source\_value | cause\_of\_death |  |  |
| cause\_source\_concept\_id | source\_concept\_id |  | From source\_to\_standard\_  concept query |

# Fact Relationship

Populate fact\_relationship from OpenMRS relationship table.

|  |  |  |  |
| --- | --- | --- | --- |
| Destination Field | Source Field | Logic | Comment |
| domain\_concept\_id\_1 |  | 56 | Person |
| fact\_id\_1 | person\_a |  | Make sure person\_id exists in CDM person table. |
| domain\_concept\_id\_2 |  | 56 | Person |
| fact\_id\_2 | person\_b |  | Make sure person\_id exists in CDM person table. |
| relationship\_concept\_id | target\_concept\_id | Use the source\_to\_standard  concept query.  Join relationship\_id  to source\_concept and vocabulary\_id = ‘OpenMRS\_  Relationship’. Assign target\_concept\_id to relationship\_concept\_id. |  |

**Acknowledements**:

Members of the OHDSI and OpenMRS Team who worked on the proof of concept project

Paul Biondich

Jon Duke

Patrick Ryan

Suranga Katshurirthne

Sri Maurya Kummamuru

OpenMRS POC Project Link

<https://wiki.openmrs.org/display/projects/OpenMRS+Support+for+the+OHDSI+Project>

OpenMRS POC ETL Mapping Specifications Link

<https://wiki.openmrs.org/display/projects/OpenMRS+Support+for+the+OHDSI+Project?preview=/84476300/86640154/OpenMRS%20ETL.docx>