Janssen Research & Development, Pharmaceutical Companies   
of Johnson & Johnson   
Common Data Model (CDM v5.0)

ETL Mapping Specification for Optum Extended SES & Extended DOD (Optum v6.0)

Qianli Ma, Erica Voss, Chris Knoll, Ajit Londhe

Version 1.0

**Table of Contents**

[1 Introduction 3](#_Toc444694120)

[1.1 Abbreviations 3](#_Toc444694121)

[1.2 Conventions Used in Document 3](#_Toc444694122)

[1.3 Processing Sequence Map 4](#_Toc444694123)

[2 Data Mapping 6](#_Toc444694124)

[2.1 Table Name: LOCATION 6](#_Toc444694125)

[2.2 Table Name: PERSON 7](#_Toc444694126)

[2.3 Table Name: OBSERVATION\_PERIOD 9](#_Toc444694127)

[2.4 Table Name: PAYER\_PLAN\_PERIOD 10](#_Toc444694128)

[2.5 Table Name: PROVIDER 12](#_Toc444694129)

[2.6 Table Name: VISIT\_OCCURRENCE 14](#_Toc444694130)

[2.7 Table Name: CONDITION\_OCCURRENCE 17](#_Toc444694131)

[2.8 Table Name: CONDITION\_ERA 21](#_Toc444694132)

[2.9 Table Name: PROCEDURE\_OCCURRENCE 22](#_Toc444694133)

[2.10 Table Name: PROCEDURE\_COST 26](#_Toc444694134)

[2.11 Table Name: DEATH 28](#_Toc444694135)

[2.12 Table Name: DRUG\_EXPOSURE 30](#_Toc444694136)

[2.13 Table Name: DRUG\_COST 33](#_Toc444694137)

[2.14 Table Name: DRUG\_ERA 34](#_Toc444694138)

[2.15 Table Name: OBSERVATION 37](#_Toc444694139)

[2.16 Table Name: CARE\_SITE 41](#_Toc444694140)

[2.17 Table Name: COHORT 42](#_Toc444694141)

[2.18 Table Name: COHORT\_ATTRIBUTE 42](#_Toc444694142)

[2.19 Table Name: MEASUREMENT 42](#_Toc444694143)

[2.20 Table Name: SPECIMEN 45](#_Toc444694144)

[2.21 Table Name: DEVICE\_EXPOSURE 46](#_Toc444694145)

[2.22 Table Name: NOTE 48](#_Toc444694146)

[2.23 Table Name: DEVICE\_COST 49](#_Toc444694147)

[2.24 Table Name: DOSE\_ERA 49](#_Toc444694148)

[2.25 Table Name: FACT\_RELATIONSHIP 50](#_Toc444694149)

[2.26 Table Name: CDM\_SOURCE 51](#_Toc444694150)

[2.27 Table Name: VISIT\_COST 52](#_Toc444694151)

[3 Code Snippets 53](#_Toc444694152)

[3.1 Vocabulary Mapping 53](#_Toc444694153)

[3.2 Domain IDs 55](#_Toc444694154)

# Introduction

The purpose of this document is to describe the Extract, Transform, Load (ETL) mapping of the licensed data from Optum Extended SES and Optum Extended DOD into the Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM) V5. Due to usage policy, the Optum ClinFormatics Extended Data Mart (CEDM) has been split into three databases: DOD (Date of Death), Other (Non-Affiliate View), and SES (Socio-Economic Status) – and the data should not be combinded into one CDM. CDM\_BUILDER will treat building SES and DOD the same for the claims information however DOD has more granularity on location and death while SES include more information on socio-economic status.

## Abbreviations

|  |  |
| --- | --- |
| **Table 1: Abbreviations** | |
| **Abbreviation** | **Description** |
| ETL | Extract, Transform, Load |
| OMOP | Observational Medical Outcomes Partnership |
| CDM | Common Data Model |
| FIPS | Federal Information Processing Standard |
| ICD | The International Classification of Diseases |
| CPT | Current Procedural Terminology |
| HCPCS | Healthcare Common Procedure Coding System |
| DRG | Diagnosis-Related Groups |
| MDC | Major Diagnostic Categories |
| CEDM | ClinFormatics Extended Data Mart |
| SES | Socio-Economic Status |
| DOD | Date of Death |

## Conventions Used in Document

The following conventions are used within this document:

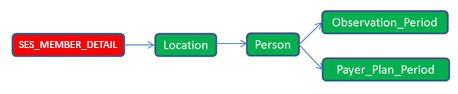
|  |  |
| --- | --- |
| **Table 2: Conventions** | |
| **Convention** | **Description** |
| - | Value does not exist |
| [X] | Value to be replaced or derived |
| *Italicized* | Referring to column in the table itself or another CDM table |
| **Bold** | Referring to a source table (from Optum Extended SES Native) |

## Processing Sequence Map

This section describes the processing sequence to build CDM tables from source data files.

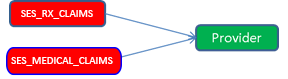
### Step I

Use **MEMBER\_DETAIL** table to build *LOCATION*, *PERSON*, *OBSERVATION\_PERIOD*, and *PAYER\_PLAN\_PERIOD* tables:



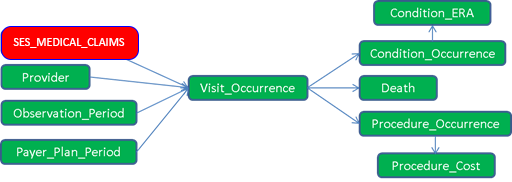
### Step II

Use **RX\_CLAIMS** and **MEDICAL\_CLAIMS** tables to build *PROVIDER* table:



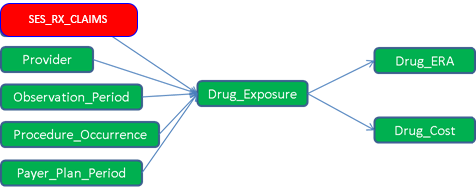
### Step III

Use *OBSERVATION\_PERIOD*, *PAYER\_PLAN\_PERIOD*, *PROVIDER* and **MEDICAL\_CLAIMS** tables to build *VISIT\_OCCURRENCE*, *CONDITION\_OCCURRENCE*, *CONDITION\_ERA*, *PROCEDURE\_OCCURRENCE*, *PROCEDURE\_COST* and *DEATH* tables:



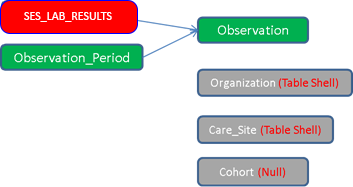
### Step IV

Use *OBSERVATION\_PERIOD*, *PAYER\_PLAN\_PERIOD*, *PROVIDER*, *PROCEDURE\_OCCURRENCE* and **RX\_CLAIMS** tables to build *DRUG\_EXPOSURE*, *DRUG\_ERA* and *DRUG\_COST* tables:



### Step V

Use *OBSERVATION\_PERIOD* and **LAB\_RESULTS** tables to build *OBSERVATION* table, and create table shells for *ORGANIZATION* and *CARE\_SITE* tables (the cohort table will not be populated at this time):



# Data Mapping

This section describes how the source files are mapped into the CDM.

The following are a list of conventions used throughout all tables in this ETL:

* Since our data is already de-identified, we do not need to mask our PERSON\_IDs.
* If a CONCEPT\_ID column cannot be mapped to a known code, it should be set to 0.
* If a non-CONCEPT\_ID column cannot be mapped to a known code, it should be set to NULL.
* Only pull records during observation periods where a person had both medical and pharmacy benefits.
* ICD diagnosis and procedure codes in **MEDICAL\_CLAIMS** table have no decimal point, so when mapping to the OMOP Vocabulary, the decimal points need to be removed from the vocabulary in order to map between the source and the vocabulary.

## Table Name: LOCATION

This table is built off the Optum **MEMBER\_DETAIL** table, which contains only census division where members live during a certain enrollment period.

Key conventions:

* Remove duplicate records before assigning LOCATION\_ID.
* In the **DOD** database use the variable **MEMBER\_DETAIL.**STATE to create the location records
* In the **SES** database use the variable **MEMBER\_DETAIL.**DIVISION to create the location records

| **Table 1: LOCATION** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| LOCATION\_ID | - | System generated. |  |
| ADDRESS\_1 | - | NULL |  |
| ADDRESS\_2 | - | NULL |  |
| CITY | - | NULL |  |
| STATE | **DOD: MEMBER\_DETAIL.**  STATE |  | Only available in DOD database, set to NULL for SES database |
| ZIP | - | NULL |  |
| COUNTY | - | NULL |  |
| LOCATION\_SOURCE\_VALUE | **SES: MEMBER\_DETAIL.**  DIVISION  **DOD**: **MEMBER\_DETAIL**.  STATE |  | In SES database use MEMBER\_DETAIL.DIVISION and in DOD database use MEMBER\_DETAIL.STATE |

## Table Name: PERSON

PERSON demographics are sourced from the **MEMBER\_DETAIL** table. All enrollment records in **MEMBER\_DETAIL** table have both medical and pharmacy coverage, but one person may have multiple records if there are gaps between enrollments. However, one record per person will be stored in the **PERSON** table.

Key conventions:

* Delete the following members: Gender changed over different enrollment period or max(YRDOB) > min(YRDOB) +2
* **MEMBER\_DETAIL** records for each person are sorted in ascending order by ELIGEFF and ELIGEND, and then use person demographics in the last record.
* After defining the patient’s YEAR\_OF\_BIRTH, delete individuals whose
  + YEAR\_OF\_BIRTH <1900 or > CURRENT\_YEAR
  + AND whose YEAR\_OF\_BIRTH > Min(Year(ELIGEFF)) +1
* For MONTH\_OF\_BIRTH, the *OBSERVATION\_PERIOD* table will need to be generated first. Then, the value can be derived from using YEAR\_OF\_BIRTH and OBSERVATION\_PERIOD\_START\_DATE
* For Race, utilize the D\_RACE\_CODE as shown below

| **Table 2: PERSON** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| PERSON\_ID | **MEMBER\_DETAIL:** PATID |  |  |
| GENDER\_CONCEPT\_ID | *GENDER\_SOURCE\_VALUE* | Map source values to their associated CONCEPT\_IDs:  M - 8507  F - 8532 | These CONCEPT\_IDs fall under VOCABULARY\_ID = ’Gender’ in CONCEPT table.  If a person has a GENDER\_SOURCE\_VALUE other than M or F they should be deleted. |
| YEAR\_OF\_BIRTH | **MEMBER\_DETAIL:** YRDOB |  |  |
| MONTH\_OF\_BIRTH | **PERSON**  YEAR\_OF\_BIRTH  **\*\*Depends on OBSERVATION\_PERIOD generation**  OBSERVATION\_PERIOD\_START\_DATE | if PERSON.YEAR\_OF\_BIRTH = YEAR(OBSERVATION\_PERIOD\_START\_DATE), then PERSON.MONTH\_OF\_BIRTH = MONTH( MIN (OBSERVATION\_PERIOD\_START\_DATE) ) | This is a derived column. Make sure to have Observation Periods generated before coming to this. |
| DAY\_OF\_BIRTH | - | NULL |  |
| RACE\_CONCEPT\_ID | **SES**  D\_RACE\_CODE | If no SES\_SES record, then map to 0.  Otherwise, map source values to their associated CONCEPT\_IDs:  W - 8527  A - 8515  U - 0  H - 0  B - 8516 | This data does not exist for DOD so this should be set to 0 for persons in the DOD database. |
| ETHNICITY\_CONCEPT\_ID | **SES**  D\_RACE\_CODE | If no SES\_SES record, then map to 0.  Otherwise, map source values to their associated CONCEPT\_IDs:  W - 0  A - 0  U - 0  H - 38003563  B - 0 | This data does not exist for DOD so this should be set to 0 for persons in the DOD database. |
| LOCATION\_ID | **SES: MEMBER\_DETAIL.**  DIVISION  **DOD: MEMBER\_DETAIL.**  STATE | **SES:** map DIVISION to LOCATION\_SOURCE\_VALUE in Location table, then extract its associated LOCATION\_ID  **DOD:** map STATE to LOCATION\_SOURCE\_VALUE in Location table, then extract its associated LOCATION\_ID |  |
| PROVIDER\_ID | - | NULL |  |
| CARE\_SITE\_ID | - | NULL |  |
| PERSON\_SOURCE\_VALUE | **MEMBER\_DETAIL**  PATID |  |  |
| GENDER\_SOURCE\_CONCEPT\_ID |  | 0 |  |
| GENDER\_SOURCE\_VALUE | **MEMBER\_DETAIL**  GDR\_CD |  |  |
| RACE\_SOURCE\_CONCEPT\_ID |  | 0 |  |
| RACE\_SOURCE\_VALUE | **SES**  D\_RACE\_CODE | If D\_RACE\_CODE in (‘W’,’A’,’U’,’B’) | This data does not exist for DOD so this should be set to NULL for persons in the DOD database. |
| ETHNICITY\_SOURCE\_CONCEPT\_ID |  | 0 |  |
| ETHNICITY\_SOURCE\_VALUE | **SES**  D\_RACE\_CODE | If D\_RACE\_CODE = ‘H’ | This data does not exist for DOD so this should be set to NULL for persons in the DOD database. |

## Table Name: OBSERVATION\_PERIOD

The **MEMBER\_DETAIL** table includes records that indicate a person’s continuous enrollment with both medical and pharmacy coverage.

Enrollment entries are consolidated by combining records that indicate continuous enrollment over a period. Consolidation is done through the following steps:

* Exclude members not included in *PERSON* table.
* **MEMBER\_DETAIL** records for each person are sorted in ascending order by ELIGEFF and ELIGEND.
* Periods of continuous enrollment are consolidated by combining records as long as the time between the end of one enrollment period and the start of the next is 32 days or less (<=32).

Key conventions:

* The gap between observation periods needs to be 32 days or less (<=32).
* Revise ELIGEFF to match the start date of Extended data (05/01/2000). Note: Depending on your organiation’s cut of data will determine what start bounds you need to enforce:

CASE

WHEN eligeff < '05/01/2000' THEN '05/01/2000'

ELSE eligeff

END

AS OBSERVATION\_PERIOD\_START\_DATE

| **Table 3: OBSERVATION\_PERIOD** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| OBSERVATION\_PERIOD\_ID | - | System generated. |  |
| PERSON\_ID | **MEMBER\_DETAIL**  PATID |  |  |
| OBSERVATION\_PERIOD\_START\_DATE | **MEMBER\_DETAIL**  ELIGEFF | Minimum start date of a continuous enrollment period. | Revise ELIGEFF using the code above to match the start date of Optum data. |
| OBSERVATION\_PERIOD\_END\_DATE | **MEMBER\_DETAIL**  ELIGEND | Maximum end date of a contiguous enrollment period. |  |
| PERIOD\_TYPE\_CONCEPT\_ID |  | 44814722-Period while enrolled in insurance |  |

## Table Name: PAYER\_PLAN\_PERIOD

For each person, create records that indicate that person’s continuous enrollment for a specific product (defined by BUS and PRODUCT) according to the **MEMBER\_DETAIL** table.

Enrollment entries are consolidated by combining records that indicate continuous enrollment over a period for a specific payer plan. Consolidation is done through the following steps:

* Exclude members not included in *PERSON* table.
* Use the combination of BUS, ASO, PRODUCT, and CDHP to define PAYER\_SOURCE\_VALUE (see table below).
* **MEMBER\_DETAIL** records for each person are sorted in ascending order by ELIGEFF, ELIGEND and PAYER\_SOURCE\_VALUE.
* Periods of continuous enrollment are consolidated by combining records as long as PAYER\_SOURCE\_VALUE and PLAN\_SOURCE\_VALUE do not change and the time between the end of one enrollment period and the start of the next is 32 days or less (<=32).

Key conventions:

* The gap between plan periods needs to be 32 days or less (<=32).
* Revise ELIGEFF to match the start date of Optum data (05/01/2000). Note: Depending on your organization’s cut of data will determine what start bounds you need to enforce.

CASE

WHEN eligeff < '05/01/2000' THEN '05/01/2000'

ELSE eligeff

END

AS PAYER\_PLAN\_PERIOD\_START\_DATE

| **Table 4: PAYER\_PLAN\_PERIOD** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| PAYER\_PLAN\_PERIOD\_ID | - | System generated. |  |
| PERSON\_ID | **MEMBER\_DETAIL:**  PATID |  |  |
| PAYER\_PLAN\_PERIOD\_START\_DATE | **MEMBER\_DETAIL:**  ELIGEFF | Minimum start date of a continuous enrollment in a plan. | Revise ELIGEFF using the code above to match the start date of Optum data. |
| PAYER\_PLAN\_PERIOD\_END\_DATE | **MEMBER\_DETAIL:**  ELIGEND | Maximum end date of a continuous enrollment in a plan. |  |
| PAYER\_SOURCE\_VALUE | **MEMBER\_DETAIL:**  BUS, ASO,  PRODUCT, CDHP | Combine, in the following order, BUS, ASO, PRODUCT, and CDHP  [W] BUS  [Y] ASO  [X] PRODUCT  [Z] CDHP  Or [W] + [Y] + [X] + [Z]  For [W] Take BUS as is, if NULL set to empty string ‘’.  If ASO = Y replace [Y] with ‘(ASO)’.  ELSE replace [Y] with ‘’.  If PRODUCT = ‘HMO’ replace [X] with ‘Health Maint Org’.  If PRODUCT = ‘PPO’ replace [X] with ‘Preferred Provider Org’  If PRODUCT = ‘EPO’ replace [X] with ‘Exclusive Provider Org’  If PRODUCT = ‘IND’ replace [X] with ‘Indemnity’  If PRODUCT = ‘POS’ replace [X] with ‘Point of Service’  If PRODUCT = ‘ALL’ replace [X] with ‘National Ancillaries, All Prod’  If PRODUCT = ‘UNK’ replace [X] with ‘Unknown’  If PRODUCT = ‘OTH’ replace [X] with ‘Other’  IF PRODUCT = NULL then replace [X] with ‘’  Else [X] = PRODUCT.  If CDHP = 1 replace [Z] with ‘(HRA)’.  If CDHP = 2 replace [Z] with ‘(HSA)’.  IF CDHP IS NULL then ‘’  Else [Z] = ‘’ |  |
| PLAN\_SOURCE\_VALUE | **MEDICAL\_CLAIMS:**  PAT\_PLANID |  |  |
| FAMILY\_SOURCE\_VALUE | - | NULL |  |

## Table Name: PROVIDER

The PROVIDER table contains a list of uniquely identified health care providers (physicians). Optum does have some provider information in **RX\_CLAIMS** and **MEDICAL\_CLAIMS** tables. However, some of the providers listed by Optum may also be considered care sites or organizations. Since there is no clear way to decipher between all items identified as providers by Optum, all of them will be added to this table, regardless if they are truly organizations or care sites.

Key conventions:

* DEA and NPI are encrypted in **RX\_CLAIMS** table.
* PROV is the provider ID in **MEDICAL\_CLAIMS** table but cannot be linked to DEA or NPI, so we will keep it as PROVIDER\_SOURCE\_VALUE too.
* Provider Specialty (PROVCAT) is available in **MEDICAL\_CLAIMS** table.
  + We added mapping of Optum provider specialty to OMOP concept - VOCABULARY\_ID = ’JNJ\_OPTUM\_P\_SPCLTY’.
* Remove duplicate records before assigning PROVIDER\_ID.

| **Table 5: PROVIDER** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| PROVIDER\_ID |  | System generated. |  |
| PROVIDER\_NAME |  | NULL |  |
| NPI | NPI | Ignore blanks, H and HHHHHHHHHH. |  |
| DEA | DEA |  |  |
| SPECIALTY\_CONCEPT\_ID | *SPECIALTY\_SOURCE\_VALUE* | **RX\_CLAIMS**   **–** set as 38004514 (Unknown Physician Specialty)  **MEDICAL\_CLAIMS –** PROVCAT: Map it to OMOP VOCABULARY\_ID = 'JNJ\_OPTUM\_P\_SPCLTY' | These CONCEPT\_IDs fall under VOCABULARY\_ID = 'JNJ\_OPTUM\_P\_SPCLTY'. A record of this vocabulary can be found in Appendix 3.1.  Set as 38004514 (Unknown Physician Specialty) if SPECIALTY\_SOURCE\_VALUE has missing value or is NULL or cannot be mapped. |
| CARE\_SITE\_ID | - | 0 |  |
| YEAR\_OF\_BIRTH | - | NULL |  |
| GENDER\_CONCEPT\_ID | - | 0 |  |
| PROVIDER\_SOURCE\_VALUE | **RX\_CLAIMS:**  NPI,  DEA  **MEDICAL\_CLAIMS:**  PROV | For NPI, ignore blanks, H and HHHHHHHHHH. | We pull in both DEA and NPI codes. Some NPI = DEA, however OPTUM has identified that as data coming in with the codes switched and not a duplication of NPI and DEA codes. |
| SPECIALTY\_SOURCE\_VALUE | **RX\_CLAIMS:** NULL  **MEDICAL\_CLAIMS:**  PROVCAT |  |  |
| SPECIALTY\_SOURCE\_CONCEPT\_ID | **-** | 0 |  |
| GENDER\_SOURCE\_VALUE | **-** | NULL |  |
| GENDER\_SOURCE\_CONCEPT\_ID | **-** | 0 |  |

## Table Name: VISIT\_OCCURRENCE

The VISIT\_OCCURRENCE table contains all person visits to health care providers, including inpatient, outpatient and ER visits. A visit is an encounter for a patient at a point of care for duration of time. There could be several providers involved in the patient's care during the Visit.

The logic derived to define inpatient visits versus emergency room visits was obtained from the following reference:  
*Scerbo, M., C. Dickstein, and A. Wilson, Health Care Data and SAS. 2001, Cary, NC: SAS Institute Inc.*

Key conventions:

* **Extract records within OBSERVATION\_PERIODs where a person has both prescription benefits and medical benefits from MEDICAL\_CLAIMS table**:
* If LST\_DT is NULL or < FST\_DT, set LST\_DT = FST\_DT.
* Cap data to fall between OBSERVATION\_PERIODS. If the OBSERVATION\_PERIOD\_START\_DATE falls between FST\_DT and LST\_DT, set FST\_DT = OBSERVATION\_PERIOD\_START\_DATE; if OBSERVATION\_PERIOD\_END\_DATE falls between FST\_DT and LST\_DT, set LST\_DT = OBSERVATION\_PERIOD\_END\_DATE. Then extract all records with both FST\_DT and LST\_DT falling between a person’s OBSERVATION\_PERIOD\_START\_DATE and OBSERVATION\_PERIOD\_END\_DATE.
* After eligible records are extracted and truncated as mentioned above, use the following steps to define visit and type of visit:

1. For each line of claim, define claim type using the following logic:

IF (RVNU\_CD >= '0100' AND RVNU\_CD <='0219') /\*Room and Board Charges\*/

OR (RVNU\_CD >= '0720' AND RVNU\_CD <='0729') /\*Labor Room and Delivery\*/

OR (RVNU\_CD >= '0800' AND RVNU\_CD <='0809') /\*Inpatient Renal Dialysis\*/

THEN

IF POS IN (13,31,32,34) THEN CLAIM\_TYPE = 'LTC'

ELSE CLAIM\_TYPE = 'IP';

ELSE IF POS IN (23)

OR (RVNU\_CD >= '0450' AND RVNU\_CD <='0459')

OR RVNU\_CD ='0981'

OR PROC\_CD IN ('99281','99282','99283','99284','99285')

THEN CLAIM\_TYPE= 'ER';

ELSE CLAIM\_TYPE = 'OP';

The LTC codes are Place of Service Codes from CMS.

\*\*This **TEMP\_MEDICAL** table can be used throughout the loading process. Pulling the following columns will help you throughout the build:

|  |  |
| --- | --- |
| * RVNU\_CD * POS * DIAG1 – DIAG5 * ICD\_FLAG * PROC1 – PROC3 * DRG * PROC\_CD * COINS | * COPAY * DEDUCT * STD\_COST * DSTATUS * UNITS * NDC * PATID * PAT\_PLANID * FST\_DT * AMT\_COB\_SAV |

1. For lines of claim with CLAIM\_TYPE = 'IP' (inpatient):
   1. Sort by PAT\_PLANID, FST\_DT, LST\_DT, PROV, and PROVCAT in ascending order.
   2. For each PAT\_PLANID, collapse lines of claim as long as the time between the LST\_DT of one line and the FST\_DT of the next is <=1 day. Then each consolidated inpatient claim is considered as one inpatient visit, and set min(FST\_DT) as VISIT\_START\_DATE, max(LST\_DT) as VISIT\_END\_DATE, ‘IP’ as PLACE\_OF\_SERVICE\_SOURCE\_VALUE.
   3. For each inpatient visit, set PROV and PROVCAT from the first claim line (all claim lines of each visit already sorted in step a) as NEW\_PROV and NEW\_PROVCAT respectively
   4. Assign a VISIT\_OCCURRENCE\_ID for each inpatient visit. Also apply this VISIT\_OCCURRENCE\_ID to the table created in Step (1) for tracking purposes.
2. Any outpatient, long term care, or emergency visits during an inpatient stay should be consolidated with that inpatient visit (i.e. if you are already in the hospital as an inpatient you most likely did not leave to go to an emergency room or outpatient visit, these records appear due to charge back reasons). The only records you do not consolidate are emergency room visits that occur on the first day of the inpatient stay (both FST\_DT and LST\_DT are equal to the VISIT\_START\_DATE of inpatient visit), this most likely is a patient who came in through the ER and later was transferred to an inpatient stay.
3. Following Step (3), for all lines of claim with CLAIM\_TYPE = 'ER' but not collapsed into inpatient visits, sort them by PAT\_PLANID, FST\_DT, LST\_DT, PROV, PROVCAT in ascending order. For each PAT\_PLANID, collapse lines of claim with the same FST\_DT in ER table as one unique ER visit, and set FST\_DT as VISIT\_START\_DATE, max (LST\_DT) as VISIT\_END\_DATE, ‘ER’ as PLACE\_OF\_SERVICE\_SOURCE\_VALUE. For each ER visit, set NEW\_PROV and NEW\_PROVCAT from the first claim line as NEW\_PROV and NEW\_PROVCAT respectively and assign VISIT\_OCCURRENCE\_ID.
4. Following Step (3), for all lines of claim with CLAIM\_TYPE = 'OP' but not collapsed into inpatient visits, sort them by PAT\_PLANID, FST\_DT, PROV, LST\_DT, PROVCAT in ascending order. For each PAT\_PLANID, collapse lines of claim with the same FST\_DT, PROV table as one unique OP visit, and set FST\_DT as VISIT\_START\_DATE, max (LST\_DT) as VISIT\_END\_DATE, ‘OP’ as PLACE\_OF\_SERVICE\_SOURCE\_VALUE. For each OP visit, set NEW\_PROV and NEW\_PROVCAT from the first claim line as NEW\_PROV and NEW\_PROVCAT respectively and assign VISIT\_OCCURRENCE\_ID.
5. After Step 2 through 5, all lines of claim in the table created in Step (1) will be assigned to a visit. Add assigned VISIT\_OCCURRENCE\_ID, NEW\_PROV, NEW\_PROVCAT, VISIT\_START\_DATE, VISIT\_END\_DATE and PLACE\_OF\_SERVICE\_SOURCE\_VALUE to each line of this table and create the **TEMP\_MEDICAL** table. The **TEMP\_MEDICAL** can be used to define *CONDITION\_OCCURRENCE*, *PROCEDURE\_OCCURRENCE*, *PROCEDURE\_COST* and *DEATH*.

* Two examples for defining *VISIT\_OCCURRENCE* can be found here:



| **Table 6: VISIT\_OCCURRENCE** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| VISIT\_OCCURRENCE\_ID | - | System generated. |  |
| PERSON\_ID | **TEMP\_MEDICAL:** PATID/PAT\_PLANID | At the row level we work with PAT\_PLANID, but PATID is what is written to the CDM. |  |
| VISIT\_START\_DATE | **TEMP\_MEDICAL:** min(FST\_DT) |  |  |
| VISIT\_START\_TIME |  | NULL |  |
| VISIT\_END\_DATE | **TEMP\_MEDICAL:** max(LST\_DT) |  |  |
| VISIT\_END\_TIME | **-** | NULL |  |
| VISIT\_TYPE\_CONCEPT\_ID | **-** | All rows will have the CONCEPT\_ID 44818517 | CONCEPT\_ID 44818517 = ‘Visit derived from encounter on claim’ |
| VISIT\_CONCEPT\_ID | **TEMP\_MEDICAL:** POS RVNU\_CD PROC\_CD | Use the steps mentioned above to create PLACE\_OF\_SERVICE\_SOURCE\_VALUE, then map it to its associated CONCEPT\_ID:  IP - 9201  OP - 9202  ER - 9203  LTC - 42898160 | These CONCEPT\_IDs fall under VOCABULARY\_ID = 'Visit' in CONCEPT table. |
| PROVIDER\_ID | **TEMP\_MEDICAL:**  NEW\_PROV + NEW\_PROVCAT | Lookup Provider ID based on NEW\_PROV and NEW\_PROVCAT |  |
| CARE\_SITE\_ID | - | NULL |  |
| VISIT\_SOURCE\_VALUE | **TEMP\_MEDICAL:** POS RVNU\_CD PROC\_CD | Use the steps mentioned above to create this field. |  |
| VISIT\_SOURCE\_CONCEPT\_ID | **-** | 0 |  |

## Table Name: CONDITION\_OCCURRENCE

Condition data can be found in the medical claim table and should be created when building VISIT\_OCCURRENCE table.

Key conventions:

* Starts from the **TEMP\_MEDICAL** table generated throughout the VISIT\_OCCURRENCE creation.
* Remove duplicates existing from DIAG1 through DIAG5: if same ICD code exists in more than one position, keep only the one with lowest position number (e.g. if 30750 exists in DIAG1 and DIAG2, then keep only 30750 in DIAG1).
* The ICD\_FLAG field defines if the claim is using ICD9 or ICD10.
  1. If ICD\_FLAG is 9, then use VOCABULARY\_ID = ‘ICD9CM’
  2. If ICD\_FLAG is 10, then use VOCABULARY\_ID = ‘ICD10CM’
* Optum removes decimal points from ICD diagnoses so when mapping to the OMOP Vocabulary, the decimal points need to also be removed from the vocabulary in order to map between the source and the vocabulary.
* Always assign its associated VISIT\_START\_DATE as CONDITION\_START\_DATE, and use NEW\_PROV and NEW\_PROVCAT (refer to VISIT\_OCCURRENCE section) to extract its ASSOCIATED\_PROVIDER\_ID from the *PROVIDER* table.
* Remove duplicate records before assigning CONDITION\_OCCURRENCE\_ID.

| **Table 7: CONDITION\_OCCURRENCE** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| CONDITION\_OCCURRENCE\_ID | - | System generated. |  |
| PERSON\_ID | **TEMP\_MEDICAL:** PATID/PAT\_PLANID | At the row level we work with PAT\_PLANID, but PATID is what is written to the CDM. |  |
| CONDITION\_CONCEPT\_ID | **TEMP\_MEDICAL:** DIAG1, DIAG2,  DIAG3,  DIAG4,  DIAG5, PROC\_CD, PROC1-3 | **DIAG1-DIAG5:**  Use Section 3.1.2 and filter with Section 4.1  **PROC1-PROC3, PROC\_CD:**  Use Section 3.1.2 and filter with Section 4.2 | As data is being assigned a CONCEPT\_ID check the domain. Only if the DOMAIN\_ID = “Condition” or if the CONCEPT\_ID = 0 can it land in the CONDITION\_OCCURRENCE table.  Other domains will need to go to their respective tables. For PROCEDURE\_OCCURRENCE and DRUG\_EXPOSURE treat the DX as an incoming procedure or drug with the lowest priority. MEASUREMENT and OBSERVATION data is described in more detail in their respective sections. |
| CONDITION\_START\_DATE | **VISIT\_OCCURRENCE:** VISIT\_START\_DATE |  |  |
| CONDITION\_END\_DATE | **-** | NULL |  |
| CONDITION\_TYPE\_CONCEPT\_ID | **VISIT\_OCCURRENCE:** PLACE\_OF\_SERVICE\_SOURCE\_VALUE | **DIAG1-5:**  Use Section 5.1  **PROC1-3, PROC\_CD:**  Use Section 5.2 | If same ICD code exists in more than one position, keep only the one with lowest position number.  Map to inpatient/outpatient header to be consistent with PROCEDURE\_OCCURRENCE table.  These CONCEPT\_IDs fall under VOCABULARY\_ID = ’Condition Type’ in CONCEPT table. |
| STOP\_REASON | - | NULL |  |
| PROVIDER\_ID | **TEMP\_MEDICAL:**  NEW\_PROV, NEW\_PROVCAT | Map NEW\_PROV and NEW\_PROVCAT to PROVIDER\_SOURCE\_VALUE and SPECIALTY\_SOURCE\_VALUE in Provider table to extract its associated Provider ID. |  |
| VISIT\_OCCURRENCE\_ID | **VISIT\_OCCURRENCE:**VISIT\_OCCURRENCE\_ID |  |  |
| CONDITION\_SOURCE\_VALUE | **TEMP\_MEDICAL:**  DIAG1, DIAG2,  DIAG3,  DIAG4,  DIAG5, PROC\_CD, PROC1-3 | Exclude records with invalid ICD9 or ICD10 diagnosis codes. | Use rules mentioned above to exclude invalid ICD9 or ICD10 diagnosis codes. |
| CONDITION\_SOURCE\_CONCEPT\_ID |  | DIAG1-5:  Use Section 3.1.1 and filter with Section 6.1  PROC\_CD, PROC1-3:  Use Section 3.1.1 and filter with Section 6.2 |  |

## Table Name: CONDITION\_ERA

*CONDITION\_ERA*s are chronological periods of condition occurrence. There will only be one type of persistence window (duration that is allowed to elapse between condition occurrences) applied to this CDM, which is 30 days.

Key conventions:

* The *CONDITION\_ERA*s are not aggregated to a higher-level class, therefore the CONDITION\_CONCEPT\_ID is unchanged.
* Use the following steps to build this table off *CONDITION\_OCCURRENCE* table:

1. Exclude records with CONDITION\_CONCEPT\_ID = 0.
2. Set CONDITION\_END\_DATE = CONDITION\_START\_DATE.
3. Sort *CONDITION\_OCCURRENCE* table by PERSON\_ID, CONDITION\_CONCEPT\_ID and CONDITION\_START\_DATE in ascending order.
4. Combine records as long as both PERSON\_ID and CONDITION\_CONCEPT\_ID don’t change and the time between CONDITION\_END\_DATE of one record and CONDITION\_START\_DATE of the next is 30 days or less (<=30).

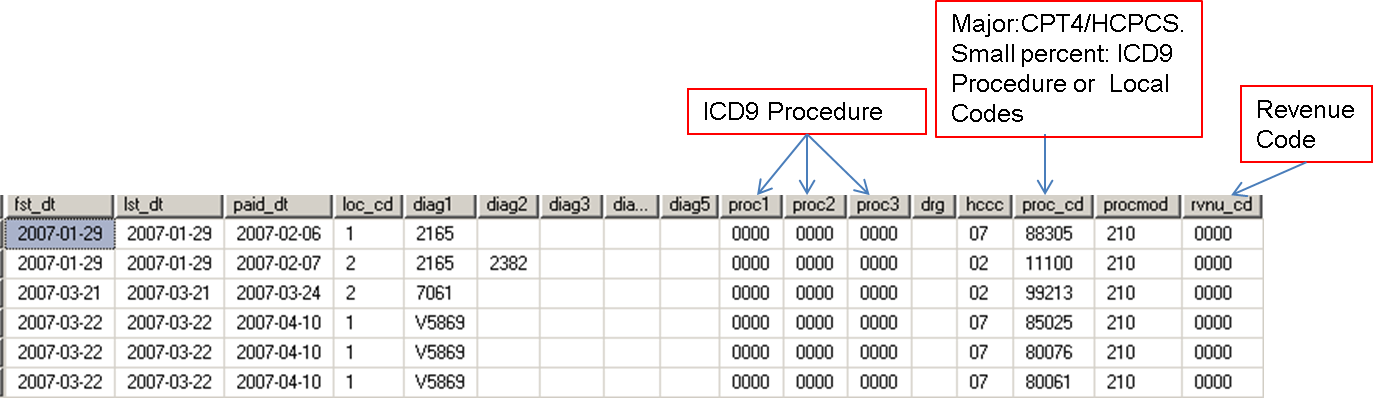
| **Table 8: CONDITION\_ERA** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| CONDITION\_ERA\_ID | - | System generated. |  |
| PERSON\_ID | **CONDITION\_OCCURRENCE:**  PERSON\_ID |  |  |
| CONDITION\_CONCEPT\_ID | **CONDITION\_OCCURRENCE:**  CONDITION\_CONCEPT\_ID | Do not build CONDITION\_ERAS where the CONDITION\_OCCURRENCE.CONDITION\_CONCEPT\_ID is 0. |  |
| CONDITION\_ERA\_START\_DATE | **CONDITION\_OCCURRENCE:**  min(CONDITION\_START\_DATE) | The start date for the condition era constructed from the individual instances of condition occurrences. It is the start date of the very first chronologically recorded instance of the condition. |  |
| CONDITION\_ERA\_END\_DATE | - | The end date for the condition era constructed from the individual instances of condition occurrences. It is the end date of the final continuously recorded instance of the condition. |  |
| CONDITION\_OCCURRENCE\_COUNT | - | Sum up the number of CONDITION\_OCCURRENCEs for this PERSON\_ID and this CONCEPT\_ID during the exposure window being built. |  |

## Table Name: PROCEDURE\_OCCURRENCE

Procedure data can be extracted from the Optum **TEMP\_MEDICAL** table and should be created when building *VISIT\_OCCURRENCE*.

Key conventions:

* Starts from the **TEMP\_MEDICAL** table generated throughout the *VISIT\_OCCURRENCE* creation.
* The following graph shows how procedure data is stored in **MEDICAL\_CLAIMS** table



* For ICD procedure codes stored in PROC1, PROC2 and PROC3:

1. ‘0’, ‘00’, ‘000’, ‘0000’, ‘00000’, ‘000000’, ‘0000000’ are usually used as place holder and will be excluded.
2. ICD9 procedure codes should contains only 2, 3 or 4 digits numeric codes (without their decimals). Exclude codes that are not in this format.
3. ICD10 procedure codes should contain 7 characters. Exclude codes that are not in this format.
4. Remove duplicates existing from PROC1 through PROC3: if same ICD procedure code exists in more than one position, keep only the one with low position number (e.g. if 7061 exists in PROC1 and PROC2, then keep only 7061 in PROC1). Then store the ICD procedure code as PROCEDURE\_SOURCE\_VALUE.
5. Following step 1 through 4, sort data by VISIT\_OCCURRENCE\_ID, PROCEDURE\_SOURCE\_VALUE and DIAG1 in ascending order. For each PROCEDURE\_SOURCE\_VALUE in a visit, assign its associated VISIT\_END\_DATE as PROCEDURE\_DATE, use NEW\_PROV and NEW\_PROVCAT (refer to VISIT\_OCCURRENCE section) extract its ASSOCIATED\_PROVIDER\_ID from *PROVIDER* table, and choose the concept\_id associated with DIAG1 in the first record as RELEVANT\_CONDITION\_CONCEPT\_ID.
6. Map PROCEDURE\_SOURCE\_VALUE to **ICD** procedure codes using code from Section 3.1.2..

* For procedure codes stored in PROC\_CD:

1. **Records with unmapped or missing values will be kept to link cost information**. Set PROC\_CD as PROCEDURE\_SOURCE\_VALUE and FST\_DT as PROCEDURE\_DATE. For each row in **TEMP\_MEDICAL** table, assign a unique PROCEDURE\_COST\_ID which will be used for creating *PROCEDURE\_COST* table.
2. Following step 1, sort data by VISIT\_OCCURRENCE\_ID, PROCEDURE\_SOURCE\_VALUE, PROCEDURE\_DATE, DIAG1, PROV, and PROVCAT. For each PROCEDURE\_SOURCE\_VALUE in a visit on the same PROCEDURE\_DATE, use NEW\_PROV and NEW\_PROVCAT from the first record to extract its ASSOCIATED\_PROVIDER\_ID from PROVIDER table, and choose the concept\_id associated with DIAG1 in the first record as RELEVANT\_CONDITION\_CONCEPT\_ID.
3. Map PROCEDURE\_SOURCE\_VALUE to CPT4/HCPCS/**ICD** procedure codes using Section 3.1.2 code.

* Remove duplicate records before assigning PROCEDURE\_OCCURRENCE\_ID.

| **Table 9: PROCEDURE\_OCCURRENCE** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| PROCEDURE\_OCCURRENCE\_ID | - | System generated. |  |
| PERSON\_ID | **TEMP\_MEDICAL:** PATID/PAT\_PLANID  **LAB\_RESULTS:**  PATID/PAT\_PLANID | At the row level we work with PAT\_PLANID, but PATID is what is written to the CDM. |  |
| PROCEDURE\_CONCEPT\_ID | **TEMP\_MEDICAL:** PROC1,  PROC2,  PROC3, PROC\_CD,  DIAG1-5  LAB\_RESULTS: LOINC, PROC\_CD | **TEMP\_MEDICAL (DIAG1-DIAG5):**  Use Section 3.1.2 and filter with Section 4.1  **TEMP\_MEDICAL (PROC1-PROC3, PROC\_CD):**  Use Section 3.1.2 and filter with Section 4.2  **LAB\_RESULTS:**  (First, look for LOINC\_CD. If no map exists, then look for PROC\_CD):  LOINC\_CD:  WHERE SOURCE\_VOCABULARY\_ID IN ('LOINC')  AND TARGET\_STANDARD\_CONCEPT IS NOT NULL  PROC\_CD:  Use Section 3.1.2 and filter with Section 4.3 | As data is being assigned a CONCEPT\_ID check the domain. While all PROC\_CD records write to the PROCEDURE\_OCCURRENCE to help the PROCEDURE\_COST table CONCEPTs that belong to other domains should land in their appropriate table. Codes that move to another table based on domain should have PROCEDURE\_CONCEPT\_ID set to 0 in the PROCEDURE\_OCCURRENCE table.  Procedures included in the DRUG\_EXPOSURE table are also included here.  0 will be set for unmapped procedure codes. |
| PROCEDURE\_DATE | **TEMP\_MEDICAL:**  FST\_DT (only for procedure code extracted from PROC\_CD)  **VISIT\_OCCURRENCE:**VISIT\_END\_DATE | For ICD procedure codes extracted from PROC1, PROC2, PROC3: use VISIT\_END\_DATE of their associated VISIT\_OCCURRENCE;  For procedure code extracted from PROC\_CD, use FST\_DT. |  |
| PROCEDURE\_TYPE\_CONCEPT\_ID | **VISIT\_OCCURRENCE:** PLACE\_OF\_SERVICE\_SOURCE\_VALUE | **TEMP\_MEDICAL (DIAG1-5):**  Use Section 5.1  **TEMP\_MEDICAL (PROC1-3, PROC\_CD):**  Use Section 5.2  **LAB\_RESULTS** table: This will be 44818702 for all rows (Lab Result) |  |
| MODIFIER\_CONCEPT\_ID | **-** | 0 |  |
| QUANTITY | **TEMP\_MEDICAL:**  Units |  |  |
| PROVIDER\_ID | **TEMP\_MEDICAL:**  NEW\_PROV,  NEW\_PROVCAT | Map NEW\_PROV and NEW\_PROVCAT to PROVIDER\_SOURCE\_VALUE and SPECIALTY\_SOURCE\_VALUE in Provider table to extract its associated Provider ID. |  |
| VISIT\_OCCURRENCE\_ID | **VISIT\_OCCURRENCE:** VISIT\_OCCURRENCE\_ID |  |  |
| PROCEDURE\_SOURCE\_VALUE | **TEMP\_MEDICAL:** PROC1,  PROC2,  PROC3, PROC\_CD,DIAG1-5  LAB\_RESULTS: PROC\_CD | Clean codes from PROC1, PROC2, PROC3 | See logic above for how to clean ICD procedure codes. |

|  |  |  |  |
| --- | --- | --- | --- |
| PROCEDURE\_SOURCE\_CONCEPT\_ID | **TEMP\_MEDICAL**  PROC1-PROC3, PROC\_CD, DIAG1-DIAG5  **LAB\_RESULTS:** PROC\_CD | **TEMP\_MEDICAL (PROC\_CD, PROC1-3):**  Use Section 3.1.1 and filter with Section 6.2  **TEMP\_MEDICAL (DIAG1-5):**  Use Section 3.1.1 and filter with Section 6.1  **LAB\_RESULTS**  Use Section 3.1.1 and filter with Section 6.3 |  |
| QUALIFIER\_SOURCE\_VALUE | **-** | NULL |  |

## Table Name: PROCEDURE\_COST

The *PROCEDURE\_COST* table captures the cost of a procedure performed on a person as well as other costs (e.g. room charge etc.). *PROCEDURE\_COST* should be loaded when building *PROCEDURE\_OCCURRENCE* table.

Key conventions:

* Costs are actually associated to the PROC\_CDs from the *PROCEDURE\_OCCURRENCE* table, they are not really associated to the PROC1 through PROC3 codes.
* We do not have enough cost information for calculating TOTAL\_PAID in Optum, use the STD\_COST instead. STD\_COST is record NULL for STD\_COST when the map does not exist.
* ‘0000’ is used as place holder for RVNU\_CD and will be set as NULL

* Since the amount of observation time in OBSERVATION\_PERIOD may be greater than that in PAYER\_PLAN\_PERIOD table, use left join to avoid excluding records when pulling PAYER\_PLAN\_PERIOD\_ID. For those records fall out of PAYER\_PLAN\_PERIOD\_START\_DATE AND PAYER\_PLAN\_PERIOD\_END\_DATE, set PAYER\_PLAN\_PERIOD\_ID as NULL.
* **Each line of claim in TEMP\_MEDICAL table will create one and only one record in PROCEDURE\_COST table.** The PROCEDURE\_COST\_ID is already created when extracting procedure codes stored in PROC\_CD for PROCEDURE\_OCCURRENCE table, thus can be easily linked to its associated PROCEDURE\_OCCURRENCE\_ID.
* Each *PROCEDURE\_OCCURRENCE* may have any number of corresponding records in the PROCEDURE\_COST table, but typically it is none (cost data not captured) or one (one payment per procedure).

| **Table 10: PROCEDURE\_COST** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| PROCEDURE\_COST\_ID | - | A system-generated unique identifier for each procedure cost record. |  |
| PROCEDURE\_OCCURRENCE\_ID | **PROCEDURE\_OCCURRENCE**: PROCEDURE\_OCCURRENCE\_ID | Same as PROCEDURE\_OCCURRENCE\_ID |  |
| CURRENCY\_CONCEPT\_ID | **-** | This will be ‘44818668’ for all rows since this is a US claims database and paid in US Dollars |  |
| PAID\_COPAY | **TEMP\_MEDICAL:**  COPAY |  |  |
| PAID\_COINSURANCE | **TEMP\_MEDICAL:**  COINS |  |  |
| PAID\_TOWARD\_DEDUCTIBLE | **TEMP\_MEDICAL:**  DEDUCT |  |  |
| PAID\_BY\_PAYER | - | NULL |  |
| PAID\_BY\_COORDINATION\_BENEFITS | **TEMP\_MEDICAL:**  AMT\_COB\_SAV |  |  |
| TOTAL\_OUT\_OF\_POCKET | **TEMP\_MEDICAL**:  COINS+ DEDUCT |  |  |
| TOTAL\_PAID | **TEMP\_MEDICAL:**  STD\_COST | NULL when STD\_COST is NULL |  |
| REVENUE\_CODE\_CONCEPT\_ID | **TEMP\_MEDICAL:**  RVNU\_CD | Use Vocabulary map 3.1.1.Filters:  WHERE SOURCE\_VOCABULARY\_ID IN ('Revenue Code')  AND TARGET\_VOCABULARY\_ID IN ('Revenue Code') |  |
| PAYER\_PLAN\_PERIOD\_ID | - | Look up associated PAYER\_PLAN\_PERIOD\_ID by PERSON\_ID, PAT\_PLANID and PROCEDURE\_DATE. If there no match, put NULL. |  |
| REVENUE\_CODE\_SOURCE\_VALUE | **TEMP\_MEDICAL:** RVNU\_CD | ‘0000’ is used as place holder for RVNU\_CD and will be set as NULL |  |

## Table Name: DEATH

The *DEATH* table is designed to capture the time when a person is deceased and cause of death. Death in Optum Extended SES/DOD can be captured by discharge status “Died” or condition in one of the various diagnosis fields containing codes indicating death in the **MEDICAL\_CLAIMS** table. This table should be loaded when building *VISIT\_OCCURRENCE* table. The DOD database has additional death information sourced from Social Security Master File.

Key conventions:

* Starts from the **TEMP\_MEDICAL** table generated throughout the *VISIT\_OCCURRENCE* creation.
* The date of death will be associated to the VISIT\_END\_DATE.
* There are three ways to identify death:
  1. Discharge Status

|  |  |  |  |
| --- | --- | --- | --- |
| CODE\_TYPE | SOURCE\_CODE | SOURCE\_CODE\_DESCRIPTION | DEATH\_TYPE\_CONCEPT\_ID |
| Discharge Status | 20 | EXPIRED | 38003566 |
| Discharge Status | 21 | EXPIRED TO BE DEFINED AT STATE LEVEL | 38003566 |
| Discharge Status | 22 | EXPIRED TO BE DEFINED AT STATE LEVEL | 38003566 |
| Discharge Status | 23 | EXPIRED TO BE DEFINED AT STATE LEVEL | 38003566 |
| Discharge Status | 24 | EXPIRED TO BE DEFINED AT STATE LEVEL | 38003566 |
| Discharge Status | 25 | EXPIRED TO BE DEFINED AT STATE LEVEL | 38003566 |
| Discharge Status | 26 | EXPIRED TO BE DEFINED AT STATE LEVEL | 38003566 |
| Discharge Status | 27 | EXPIRED TO BE DEFINED AT STATE LEVEL | 38003566 |
| Discharge Status | 27 | EXPIRED TO BE DEFINED AT STATE LEVEL | 38003566 |
| Discharge Status | 28 | EXPIRED TO BE DEFINED AT STATE LEVEL | 38003566 |
| Discharge Status | 29 | EXPIRED TO BE DEFINED AT STATE LEVEL | 38003566 |
| Discharge Status | 40 | EXPIRED AT HOME (HOSPICE) | 38003566 |
| Discharge Status | 41 | EXPIRED IN A MEDICAL FACILITY (HOSPICE) | 38003566 |
| Discharge Status | 42 | EXPIRED – PLACE UNKNOWN (HOSPICE) | 38003566 |

* 1. The following file provides codes for identifying death from **TEMP\_MEDICAL** and mapping to DEATH\_TYPE\_CONCEPT\_ID. The death codes can come from ICD9s, ICD10s, discharge status, or DRGs.



* 1. ***(DOD Only)*** pulling data from the DOD\_DEATH table.
* Only one record will be kept for each patient; use the following hierarchy to determine which record to use
  1. *(****DOD Only)*** Death recorded in DOD\_DEATH table
  2. Death captured by discharge status
  3. Death captured by DRGs
  4. Death captured by ICD9 or ICD10 diagnosis codes.
     1. If ICD\_FLAG = 9 then use the ICD9 codes
     2. If ICD\_FLAG = 10 then use the ICD10 codes
* After determining which table or code the death should come from take the record with the latest date of death.
* If there are health care visits after 30 days of death date, delete the death record.

| **Table 11: DEATH** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| PERSON\_ID | **TEMP\_MEDICAL:** PATID/PAT\_PLANID  (DOD only)  **DEATH**  PATID/PAT\_PLANID | At the row level we work with PAT\_PLANID, but PATID is what is written to the CDM. |  |
| DEATH\_DATE | **VISIT\_OCCURRENCE:**  VISIT\_END\_DATE  (DOD only):  **DEATH**  ymdod |  |  |
| DEATH\_TYPE\_CONCEPT\_ID | Discharge status:  **TEMP\_MEDICAL:** DSTATUS  Condition:  **TEMP\_MEDICAL:**  ICD\_FLAG, DIAG1, DIAG2, DIAG3,DIAG4,DIAG5  DRG:  **TEMP\_MEDICAL:**  DRG  Death status (DOD only):  **DEATH** | **TEMP\_MEDICAL  (DSTATUS):**  **– Priority 2 -**  38003566  **TEMP\_MEDICAL (DIAG1-DIAG5):**  **- Priority 4 -**  38003567  **TEMP\_MEDICAL (DRG):**  **- Priority 3 -**  38003568  **DOD\_DEATH:**  **- Priority 1 -**  38003569 | These CONCEPT\_IDs fall under VOCABULARY\_ID = ’Death Type’ in CONCEPT table.  The death type for social security data is not optimal and will update if a better code becomes available. |
| CAUSE\_OF\_DEATH\_CONCEPT\_ID | - | 0 |  |
| CAUSE\_OF\_DEATH\_SOURCE\_VALUE | NULL |  |  |
| CAUSE\_SOURCE\_CONCEPT\_ID | - | 0 |  |

## Table Name: DRUG\_EXPOSURE

Drug exposures are garnered from **RX\_CLAIMS** table but can also be extracted from procedure codes found in the *PROCEDURE\_OCCURRENCE* table.

Key conventions:

* **Prescription drug**:
  + - 1. Keep records with FILL\_DT that fall within an *OBSERVATION\_PERIOD* available for each person.
      2. Remove duplicate records within each RX claim (identified by PAT\_PLANID and CLMID): Sort Rx table by PAT\_PLANID, CLMID, NDC, COPAY, DEDUCT, DISPFEE, AVGWHLSL, STD\_COST, DAYS\_SUP, QUANTITY, RFL\_NBR, MAIL\_IND, NPI and DEA in ascending order. For each PAT\_PLANID, CLMID,NDC combination, keep the last record.
      3. Following previous step, smoosh records to remove reversals: Group data by PAT\_PLANID, NDC, and FILL\_DT, then keep max(RFL\_NBR) as RFL\_NBR, max(MAIL\_IND) as MAIL\_IND. Also if NPI exists keep max(NPI) as PROVIDER, if NPI does not exist try max(DEA). Cost information will be summed up and this intermediate table will be used for populating DRUG\_COST table.

set sum (QUANTITY) as QUANTITY – rounded to a whole number (e.g. 2.6 = 3)

sum (DAYS\_SUP) as DAYS\_SUPPLY, then apply this rule: if DAYS\_SUPPLY > 365, set it as 365, if DAYS\_SUPPLY <0, set it as 0.

sum (COPAY) as PAID\_COPAY, sum(DEDUCT) as PAID\_TOWARD\_DEDUCTIBLE, sum (DISPFEE) as DISPENSING\_FEE, sum (AVGWHLSL) as AVERAGE\_WHOLESALE\_PRICE, sum (STD\_COST) as TOTAL\_PAID

| **Table 12: DRUG\_EXPOSURE** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| DRUG\_EXPOSURE\_ID | - | System generated. |  |
| PERSON\_ID | **RX\_CLAIMS:** PATID/PAT\_PLANID  **TEMP\_MEDICAL:** PATID/PAT\_PLAN\_ID | At the row level we work with PAT\_PLANID, but PATID is what is written to the CDM. |  |
| DRUG\_CONCEPT\_ID | **RX\_CLAIMS:**  NDC  TEMP\_MEDICAL:  PROC1-PROC3, PROC\_CD | **NDC:**  Use Vocab map 3.1.2.  Use filter:  WHERE SOURCE\_VOCABULARY\_ID IN ('NDC')  AND TARGET\_STANDARD\_CONCEPT= 'S'  AND TARGET\_INVALID\_REASON IS NULL  AND FILL\_DT BETWEEN SOURCE\_VALID\_START\_DATE AND SOURCE\_VALID\_END\_DATE  **TEMP\_MEDICAL (PROC1-PROC3, PROC\_CD):**  Use Section 3.1.2 and filter with Section 4.2 | For procedure drugs, the CONCEPT\_ID must have a DOMAIN\_ID of ‘Drug’ to be included. |
| DRUG\_EXPOSURE\_START\_DATE | **RX\_CLAIMS:** FILL\_DT  **TEMP\_MEDICAL: PROC1-3, PROC\_CD:** VISIT\_END\_DATE |  |  |
| DRUG\_EXPOSURE\_END\_DATE | - | NULL |  |
| DRUG\_TYPE\_CONCEPT\_ID | **RX\_CLAIMS:** MAIL\_IND  **TEMP\_MEDICAL: PROC1-3, PROC\_CD:** PLACE\_OF\_SERVICE\_SOURCE\_VALUE | **Prescription drug**:  If MAIL\_IND = ‘Y’ then 38000176 (Prescription dispensed through mail order); else 38000175 (Prescription dispensed in pharmacy)  **Procedure drug**: if  IP then 38000180 (Inpatient administration); else 38000179 (Physician administered drug, identified as procedure) | These CONCEPT\_IDs fall under VOCABULARY\_ID = ‘Drug Type’ in CONCEPT table. |
| STOP\_REASON | - | NULL |  |
| REFILLS | **RX\_CLAIMS:** RFL\_NBR | NULL for **Procedure drug** |  |
| QUANTITY | **RX\_CLAIMS:** sum(QUANTITY)    **TEMP\_MEDICAL: PROC1-3, PROC\_CD:**  sum(UNITS) |  |  |
| DAYS\_SUPPLY | **RX\_CLAIMS :**  sum(DAYS\_SUP) | NULL for **Procedure drug** |  |
| SIG | - | NULL |  |
| ROUTE\_CONCEPT\_ID | - | 0 |  |
| EFFECTIVE\_DRUG\_DOSE | - | NULL |  |
| DOSE\_UNIT\_CONCEPT\_ID | - | 0 |  |
| LOT\_NUMBER | - | NULL |  |
| PROVIDER\_ID | **RX\_CLAIMS:** NPI,  DEA  **TEMP\_MEDICAL: PROC1-3, PROC\_CD:**  NEW\_PROV,  NEW\_PROVCAT | **Prescription drug**:  Map NPI or DEA to PROVIDER\_SOURCE\_VALUE in PROVIDER table, and extract its associated PROVIDER\_ID;  **Procedure drug**:  Map NEW\_PROV and NEW\_PROVCAT to PROVIDER\_SOURCE\_VALUE and SPECIALTY\_SOURCE\_VALUE in Provider table to extract its associated Provider ID. |  |
| VISIT\_OCCURRENCE\_ID | **VISIT\_OCCURRENCE\_ID:**  VISIT\_OCCURRENCE\_ID | **Prescription drug**:  NULL  **Procedure drug**: VISIT\_OCCURRENCE\_ID |  |
| DRUG\_SOURCE\_VALUE | **RX\_CLAIMS:**  NDC  **TEMP\_MEDICAL:** PROC\_CD, PROC1-3  **SES/DOD\_LAB\_RESULTS:**  PROC\_CD |  |  |
| DRUG\_SOURCE\_CONCEPT\_ID | **RX\_CLAIMS:**  NDC  **TEMP\_MEDICAL:** PROC\_CD, PROC1-3  **SES/DOD\_LAB\_RESULTS:**  PROC\_CD | Filter when drug coming from **RX\_CLAIMS**:  WHERE SOURCE\_VOCABULARY\_ID IN ('NDC')  AND TARGET\_VOCABULARY\_ID IN ('NDC')  AND FILL\_DT BETWEEN SOURCE\_VALID\_START\_DATE AND SOURCE\_VALID\_END\_DATE  **TEMP\_MEDICAL (PROC\_CD, PROC1-3):**  Use Section 3.1.1 and filter with Section 6.2  **SES/DOD\_LAB\_RESULTS**  Use Section 3.1.1 and filter with Section 6.3 |  |
| ROUTE\_SOURCE\_VALUE | **-** | **0** |  |
| DOSE\_UNIT\_SOURCE\_VALUE | **-** | **0** |  |

## Table Name: DRUG\_COST

*DRUG\_COST* information is pulled from the **RX\_CLAIMS** table at the same time the *DRUG\_EXPOSURE* is loaded. Also, it is not appropriate to pull across procedure drug costs into this table.

Key conventions:

* Starts from the intermediate table created throughout *DRUG\_EXPOSURE* table creation (prescription drug, step 3).
* We don’t have the information for calculating TOTAL\_PAID in Optum, use the STD\_COST.
* Since the amount of observation time in OBSERVATION\_PERIOD may be greater than that in *PAYER\_PLAN\_PERIOD* table, use left join to avoid excluding records when pulling PAYER\_PLAN\_PERIOD\_ID. For those records fall out of PAYER\_PLAN\_PERIOD\_START\_DATE AND PAYER\_PLAN\_PERIOD\_END\_DATE, set PAYER\_PLAN\_PERIOD\_ID as NULL.

| **Table 13: DRUG\_COST** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| DRUG\_COST\_ID | - | Same as DRUG\_EXPOSURE\_ID | The relationship between DRUG\_COST and DRUG\_EXPOSURE is one to one. |
| DRUG\_EXPOSURE\_ID | **DRUG\_EXPOSURE:** DRUG\_EXPOSURE\_ID | Same as DRUG\_EXPOSURE\_ID |  |
| CURRENCY\_CONCEPT\_ID | **-** | This will be ‘44818668’ for all rows since this is a US claims database and paid in US Dollars |  |
| PAID\_COPAY | **RX\_CLAIMS:** sum(COPAY) |  |  |
| PAID\_COINSURANCE | - | NULL |  |
| PAID\_TOWARD\_DEDUCTIBLE | **RX\_CLAIMS:** sum(DEDUCT) |  |  |
| PAID\_BY\_PAYER | - | NULL |  |
| PAID\_BY\_COORDINATION\_BENEFITS | - | NULL |  |
| TOTAL\_OUT\_OF\_POCKET | - | NULL |  |
| TOTAL\_PAID | **RX\_CLAIMS:** sum(STD\_COST) |  |  |
| INGREDIENT\_COST | - | NULL |  |
| DISPENSING\_FEE | **RX\_CLAIMS:** sum(DISPFEE) |  |  |
| AVERAGE\_WHOLESALE\_PRICE | **RX\_CLAIMS:** sum(AVGWHLSL) |  |  |
| PAYER\_PLAN\_PERIOD\_ID | **PAYER\_PLAN\_PERIOD:** PAYER\_PLAN\_PERIOD\_ID | Look up associated PAYER\_PLAN\_PERIOD\_ID by PERSON\_ID and DRUG\_EXPOSURE\_START\_DATE AND PAT\_PLANID. If there no match, put NULL. |  |

## Table Name: DRUG\_ERA

A Drug Era is defined as a span of time when the Person is assumed to be exposed to a particular drug. Successive periods of Drug Exposures are combined under certain rules to produce continuous Drug Eras. The *DRUG\_ERA* table is populated by pulling from the *DRUG\_EXPOSURE* table within the CDM. Drug eras are consolidated to their respective ingredient off the *DRUG\_EXPOSURE* table. A drug era is therefore understood as exposure to a certain compound over a certain period of time. There will only be one type of persistence window (duration that is allowed to elapse between drug exposures) applied to this CDM, which is 30 days.

Key conventions:

* Use the following steps to build this table off *DRUG\_EXPOSURE* table:

1. Exclude records with DRUG\_CONCEPT\_ID = 0.
2. Use the following logic to map DRUG\_CONCEPT\_ID to ingredient levels:

Select distinct A.concept\_id as Drug\_EXPOSURE\_concept\_id,

C.concept\_id as ingredient\_concept\_id

FROM CONCEPT C

JOIN CONCEPT\_ANCESTOR CA

ON CA.ancestor\_concept\_id = C.concept\_id

and c.vocabulary\_id = 'RxNorm'

and c.concept\_class\_id = 'Ingredient'

and invalid\_reason is null

JOIN concept A

ON CA.descendant\_CONCEPT\_ID = A.CONCEPT\_ID

1. Replace the values of DRUG\_CONCEPT\_ID with their ingredient CONCEPT\_IDs identified by step 2), and exclude records with DRUG\_CONCEPT\_IDs that can’t be mapped to ingredient level.
2. Calculate DRUG\_EXPOSURE\_END\_DATE: If DRUG\_TYPE\_CONCEPT\_ID in (38000175, 38000176), set to DRUG\_EXPOSURE\_START\_DATE+DAYS\_SUPPLY;

Else set to DRUG\_EXPOSURE\_START\_DATE.

1. Sort *DRUG\_EXPOSURE* table by PERSON\_ID, DRUG\_CONCEPT\_ID, DRUG\_EXPOSURE\_START\_DATE and DRUG\_EXPOSURE\_END\_DATE in ascending order.
2. Combine records as long as both PERSON\_ID and DRUG\_CONCEPT\_ID don’t change and the time between DRUG\_EXPOSURE\_END\_DATE of one record and DRUG\_EXPOSURE\_START\_DATE of the next is 30 days or less (<=30).

* Compound drugs can create multiple ERAs since they can be mapped to multiple ingredients.

| **Table 14: DRUG\_ERA** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| DRUG\_ERA\_ID | - | System generated. |  |
| PERSON\_ID | **DRUG\_EXPOSURE:**  PERSON\_ID |  |  |
| DRUG\_CONCEPT\_ID | **DRUG\_EXPOSURE:** DRUG\_CONCEPT\_ID | Use the logic above to map to ingredient CONCEPT\_ID and exclude records. |  |
| DRUG\_ERA\_START\_DATE | **DRUG\_EXPOSURE:** min(DRUG\_EXPOSURE\_START\_DATE) | The start date for the drug era constructed from the individual instances of drug exposures. It is the start date of the very first chronologically recorded instance of utilization of a drug. |  |
| DRUG\_ERA\_END\_DATE | **DRUG\_EXPOSURE:**  max(DRUG\_EXPOSURE\_END\_DATE) | The end date for the drug era constructed from the individual instance of drug exposures. It is the end date of the final continuously recorded instance of utilization of a drug. |  |
| DRUG\_EXPOSURE\_COUNT | - | Sum up the number of *DRUG\_EXPOSURE*s for this PERSON\_ID and this CONCEPT\_ID during the exposure window being built. |  |
| GAP\_DAYS | - | Sum of the days in the drug\_era that were not covered by a drug\_exposure\_record |  |

## Table Name: OBSERVATION

Observations describe symptoms, clinical observations, etc. about the person.

Key conventions:

* Optum Extended SES does include a Health Risk Assessment table (SES\_HRA), but for the purposes of this document, we will not include this data at this time due to the ambiguities of the source data. Future versions may include it once clarifications on the content of that table are made available by Optum.
* Only include records with OBSERVATION\_DATE that fall within an *OBSERVATION\_PERIOD*s.

n

| **Table 15:**  **OBSERVATION** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| OBSERVATION\_ID |  | A system generated unique identifier for each observation. |  |
| PERSON\_ID | **TEMP\_MEDICAL:**  PATID/PAT\_PLANID  **LAB\_RESULTS:**  PATID / PAT\_PLANID | At the row level we work with PAT\_PLANID, but PATID is what is written to the CDM. |  |
| OBSERVATION\_CONCEPT\_ID | **TEMP\_MEDICAL:**  DIAG1-DIAG5; PROC1-PROC3, PROC\_CD,  DRG  **SES/DOD\_LAB\_RESULTS**:  LOINC\_CD  PROC\_CD | **TEMP\_MEDICAL (DIAG1-DIAG5):**  Use Section 4.1  **TEMP\_MEDICAL (PROC-PROC3, PROC\_CD):**  Use Section 3.1.2 and filter with Section 4.2  DRGs still require a mapping. DRGs change over time so we need to map to the DRG that is valid at the time the procedure occurred. Notice we do not set INVALID\_REASON IS NULL – since we are using historical codes, some DRGs we use will be invalid today but not at the time of the procedure.  Use Vocab map 3.1.2, but **do not** set INVALID\_REASON to NULL  Use filter:  WHERE SOURCE\_VOCABULARY\_ID IN ('DRG')  AND TARGET\_VOCABULARY\_ID IN ('DRG')  AND TARGET\_STANDARD\_CONCEPT IS NOT NULL  **LAB\_RESULTS:**  (First, look for LOINC\_CD. If no map exists, then look for PROC\_CD):  Use Section 3.1.2 and filter with Section 4.3 |  |
| OBSERVATION\_DATE | **TEMP\_MEDICAL:**  FST\_DT  **LAB\_RESULTS:**  FST\_DT |  |  |
| OBSERVATION\_TIME | - | NULL |  |
| VALUE\_AS\_NUMBER | **TEMP\_MEDICAL:**  UNITS  **LAB\_RESULTS**:  RSLT\_NBR |  |  |
| VALUE\_AS\_STRING | **-** | NULL |  |
| VALUE\_AS\_CONCEPT\_ID | **LAB\_RESULTS**  ABNL\_CD  RSLT\_TXT  **TEMP\_MEDICAL:**  DIAG1-DIAG5; PROC1-PROC3, PROC\_CD | LAB\_RESULTS:  Follow logic given in MEASUREMENT table.  **TEMP\_MEDICAL:**  Use 3.1.3 and point it at the SOURCE\_CODE. But only take the top TARGET\_CONCEPT\_ID because a few SOURCE\_CODES can map to multiple TARGET\_CONCEPT\_ID. |  |
| UNIT\_CONCEPT\_ID | **TEMP\_MEDICAL:** -  **LAB\_RESULTS**:  RSLT\_UNIT\_NM | Null  Filters:  WHERE SOURCE\_VOCABULARY\_ID IN ('UCUM')  AND TARGET\_VOCABULARY\_ID IN ('UCUM')  AND TARGET\_INVALID\_REASON IS NULL |  |
| RANGE\_LOW | **TEMP\_MEDICAL:** -  **LAB\_RESULTS:**  LOW\_NRML |  |  |
| RANGE\_HIGH | **TEMP\_MEDICAL:** -  **LAB\_RESULTS:**  HI\_NRML |  |  |
| OBSERVATION\_TYPE\_CONCEPT\_ID | **TEMP\_MEDICAL:**  UNITS  **LAB\_RESULTS:**  Take whatever was set in the MEASUREMENT table | **TEMP\_MEDICAL:**  DIAG1-DIAG5; PROC1-PROC3, PROC\_CD  Take the type assigned in 5.1 and 5.2.  DRG:  900000008-Claim DRG  LAB:  38000277 – Observation numeric result |  |
| ASSOCIATED\_PROVIDER\_ID | **TEMP\_MEDICAL:**  PROV | NULL |  |
| VISIT\_OCCURRENCE\_ID | **TEMP\_MEDICAL:**  VISIT\_OCCURRENCE\_ID  **LAB\_RESULTS:**  NULL | Refer to logic in building VISIT\_OCCURRENCE table for linking with VISIT\_OCCURRENCE\_ID |  |
| RELEVANT\_CONDITION\_CONCEPT\_ID | - | NULL |  |
| OBSERVATION\_SOURCE\_VALUE | **TEMP\_MEDICAL:**  DIAG1-DIAG5; PROC1-PROC3, PROC\_CD, DRG  **LAB\_RESULTS**:  LOINC\_CD  PROC\_CD | **LAB\_RESULTS**:  The LOINC\_CD or PROC\_CD as it appears in the SES/DOD\_LAB\_RESULTS table and as was selected above (i.e. use the LOINC\_CD first if possible). |  |
| UNIT\_SOURCE\_VALUE | **TEMP\_MEDICAL:**  NULL  **LAB\_RESULTS**:  RSLT\_UNIT\_NM |  |  |

## Table Name: CARE\_SITE

The *CARE\_SITE* table contains a list of uniquely identified points of care, or an individual clinical location within an organization. Each *CARE\_SITE* belongs to one *ORGANIZATION*.

Optum does not have clear care site information so this table will only contain one row representing the fact that no care site information will be captured.

| **Table 16: CARE\_SITE** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| CARE\_SITE\_ID | 0 |  |  |
| LOCATION\_ID | 0 |  |  |
| ORGANIZATION\_ID | 0 |  |  |
| PLACE\_OF\_SERVICE\_CONCEPT\_ID | 0 |  |  |
| CARE\_SITE\_SOURCE\_VALUE | - | NULL |  |
| PLACE\_OF\_SERVICE\_SOURCE\_VALUE | - | NULL |  |

## Table Name: COHORT

Cohort defines groups of entities that are exposed to common circumstances. At this time, we will not incorporate any special logic to create cohorts however in the future this logic will be provided from the epidemiology team.

| **Table 17: COHORT** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| COHORT\_ID |  |  |  |
| COHORT\_DEFINITION\_ID |  |  |  |
| COHORT\_START\_DATE |  |  |  |
| COHORT\_END\_DATE |  |  |  |
| SUBJECT\_ID |  |  |  |
| STOP\_REASON |  |  |  |

## Table Name: COHORT\_ATTRIBUTE

The *COHORT\_ATTRIBUTE* table contains attributes associated with each subject within a cohort, as defined by a given set of criteria for a duration of time.

| **Table 18: COHORT\_ATTRIBUTE** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| COHORT\_DEFINITION\_ID |  |  |  |
| SUBJECT\_ID |  |  |  |
| COHORT\_START\_DATE |  |  |  |
| COHORT\_END\_DATE |  |  |  |
| ATTRIBUTE\_DEFINITION\_ID |  |  |  |
| VALUE\_AS\_NUMBER |  |  |  |
| VALUE\_AS\_CONCEPT\_ID |  |  |  |

## Table Name: MEASUREMENT

The Measurement table contains lab tests, vital signs, etc. that were conducted during a valid *OBSERVATION\_PERIOD*. Lab values are sourced from the Optum **LAB\_RESULTS** and the **TEMP\_MEDICAL** tables. Lab Observations from **LAB\_RESULTS** are tracked using standard LOINC codes which are mapped to standard concept identifiers. Units of measure are also mapped to standard concept identifiers in the dictionary for standardization. Lab observations from **TEMP\_MEDICAL** are codes in the DIAG1-DIAG5 and PROC1-PROC3 fields that have been mapped to standard concepts with DOMAIN\_IDs of ‘Measurement’.

* Units are mapped to UNIT\_CONCEPT\_IDs in the OMOP VOCABULARY (VOCABULARY\_ID = ‘UCUM’ - Unified Code for Units of Measure (UCUM)). Please note that mapping a UNIT\_SOURCE\_VALUE to a UNIT\_CONCEPT\_ID is both **case sensitive** and **accent sensitive**.
* If both the LOINC\_CD and TST\_DESC fields are blank from **LAB\_RESULTS**, this row is not considered useful and should not be used.
* LOINCs
  1. Valid LOINC codes have the following layouts #-#, ##-#, ###-#, ####-#, and #####-# .
  2. When mapping to valid LOINCs in the OMOP Vocabulary (VOCABULARY\_ID = ‘LOINC’) there are a few invalid LOINC codes. Implementing a check for the second to last character is a ‘-‘ ensures you pull a valid LOINC from the VOCABULARY.
* Lab results in Optum **LAB\_RESULTS** is stored in three fields: ABNL\_CD, RSLT\_TXT and RSLT\_NBR. Numeric results can be in both RSLT\_TXT and RSLT\_NBR, and RSLT\_NBR usually has the following values if the result is string: **0 or -999999.999999**.
* ABNL\_CD is the abnormal indicator set by the lab vendors: **‘A’** or **‘AB’** means “abnormal”, **‘N’** means “normal”, **‘H’** means “Above the normal range”, **‘L’** means “Below the normal range”. Use the following logic to assign value to VALUE\_AS\_NUMBER, VALUE\_AS\_CONCEPT\_ID and OBSERVATION\_TYPE\_CONCEPT\_ID:

VALUE\_SOURCE\_VALUE = RSLT\_TXT;

/\*Result as concept code\*/

IF LOWER(RSLT\_TXT) ='low' OR ABNL\_CD ='L' THEN VALUE\_AS\_CONCEPT\_ID = **4267416**;

ELSE IF LOWER(RSLT\_TXT) ='high' OR ABNL\_CD ='H' THEN VALUE\_AS\_CONCEPT\_ID =**4328749**;

ELSE IF LOWER(RSLT\_TXT) ='normal' OR ABNL\_CD ='N' THEN VALUE\_AS\_CONCEPT\_ID =**4069590**;

ELSE IF LOWER(RSLT\_TXT) ='abnormal' OR ABNL\_CD IN ('A','AB')

THEN VALUE\_AS\_CONCEPT\_ID =**4135493**;

ELSE IF LOWER(RSLT\_TXT) ='absent' THEN VALUE\_AS\_CONCEPT\_ID =**4132135**;

ELSE IF LOWER(RSLT\_TXT) ='present' THEN VALUE\_AS\_CONCEPT\_ID =**4181412**;

ELSE IF LOWER(RSLT\_TXT) ='positive' THEN VALUE\_AS\_CONCEPT\_ID =**9191**;

ELSE IF LOWER(RSLT\_TXT) ='negative' THEN VALUE\_AS\_CONCEPT\_ID =**9189**;

ELSE IF LOWER(RSLT\_TXT) in ('final','final report') THEN VALUE\_AS\_CONCEPT\_ID =**9188**;

ELSE IF LOWER(RSLT\_TXT) ='not found' THEN VALUE\_AS\_CONCEPT\_ID =**9190**;

ELSE IF LOWER(RSLT\_TXT) ='trace' THEN VALUE\_AS\_CONCEPT\_ID = **9192**;

IF RSLT\_NBR NOT IN (**0**, -**999999.999999**) THEN DO; /\*Result as number\*/

VALUE\_AS\_NUMBER = RSLT\_NBR;

MEASUREMENT\_TYPE\_CONCEPT\_ID = **38000277**;

END;

ELSE IF MEASUREMENT\_CONCEPT\_ID is not null THEN

MEASUREMENT\_TYPE\_CONCEPT\_ID = **38000279**; /\*Result as Concept\*/

ELSE MEASUREMENT\_TYPE\_CONCEPT\_ID = **38000278**; /\*Result as Text\*/

* The RSLT\_TXT field tends to have operators (<, >, =, <=, >=) included in the first 2 characters of the string. Use the following logic:

case substring(rslt\_txt, 0, 2)

when '>' then operator\_concept\_id = 4172704

when '<' then operator\_concept\_id = 4171756

when '=' then operator\_concept\_id = 4172703

when '>=' then operator\_concept\_id = 4171755

when '<=' then operator\_concept\_id = 4171754

end

| **Table 19: MEASUREMENT** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| MEASUREMENT\_ID | - | System Generated |  |
| PERSON\_ID | **LAB\_RESULTS:**  PATID / PAT\_PLANID  **TEMP\_MEDICAL:**  PATID / PAT\_PLANID | At the row level we work with PAT\_PLANID, but PATID is what is written to the CDM. |  |
| MEASUREMENT\_CONCEPT\_ID | **LAB\_RESULTS**:  LOINC\_CD  PROC\_CD  **TEMP\_MEDICAL:**  DIAG1-DIAG5; PROC1-PROC3, PROC\_CD | **LAB\_RESULTS:**  (First, look for LOINC\_CD. If no map exists, then look for PROC\_CD):  Use Section 3.1.2 and filter with Section 4.3  **TEMP\_MEDICAL (DIAG1-DIAG5):**  Use Section 3.1.2 and filter with Section 4.1  **TEMP\_MEDICAL (PROC1-PROC3, PROC\_CD):**  Use Section 3.1.2 and filter with Section 4.2  From LOINC\_CD:  WHERE SOURCE\_VOCABULARY\_ID IN ('LOINC')  AND TARGET\_STANDARD\_CONCEPT IS NOT NULL |  |
| MEASUREMENT\_ DATE | **LAB\_RESULTS:**  FST\_DT  **TEMP\_MEDICAL:**  FST\_DT |  |  |
| MEASUREMENT\_ TIME | - | NULL |  |
| MEASUREMENT\_ TYPE\_CONCEPT\_ID | - | From **LAB\_RESULTS** table: This will be 44818702 for all rows (Lab Result)  If the record is coming from another table, like the CONDITION\_OCCURENCE, keep the types that would have been assigned in that table. |  |
| OPERATOR\_ CONCEPT\_ID | **LAB\_RESULTS:**  RSLT\_TXT | Use first 2 characters of RSLT\_TXT as described above to get the concept ID |  |
| VALUE\_AS\_NUMBER | **LAB\_RESULTS**:  RSLT\_NBR  **TEMP\_MEDICAL:**  UNITS |  |  |
| VALUE\_AS\_ CONCEPT\_ID | **LAB\_RESULTS**  ABNL\_CD  RSLT\_TXT  **TEMP\_MEDICAL:**  DIAG1-DIAG5; PROC1-PROC3, PROC\_CD | **LAB\_RESULTS**  Refer to logic above  **TEMP\_MEDICAL:**  Use 3.1.3 and point it at the SOURCE\_CODE. But only take the top TARGET\_CONCEPT\_ID because a few SOURCE\_CODES can map to multiple TARGET\_CONCEPT\_ID. |  |
| UNIT\_CONCEPT\_ID | **LAB\_RESULTS**:  RSLT\_UNIT\_NM | Filters:  WHERE SOURCE\_VOCABULARY\_ID IN ('UCUM')  AND TARGET\_VOCABULARY\_ID IN ('UCUM')  AND TARGET\_INVALID\_REASON IS NULL | Use the code in Section 3.1.1.  This vocabulary is case sensitive. |
| RANGE\_LOW | **LAB\_RESULTS:**  LOW\_NRML |  |  |
| RANGE\_HIGH | **LAB\_RESULTS:**  HI\_NRML |  |  |
| PROVIDER\_ID | **TEMP\_MEDICAL:**  PROV |  |  |
| VISIT\_OCCURRENCE\_ID | **Measurement from TEMP\_MEDICAL tables use VISIT\_OCCURRENCE**: VISIT\_OCCURRENCE\_ID | Refer to logic in building VISIT\_OCCURRENCE table for linking with VISIT\_OCCURRENCE\_ID |  |
| MEASUREMENT\_ SOURCE\_VALUE | **LAB\_RESULTS**:  LOINC\_CD  PROC\_CD  **TEMP\_MEDICAL:**  DIAG1-DIAG5; PROC1-PROC3, PROC\_CD | **LAB\_RESULTS**:  The LOINC\_CD or PROC\_CD as it appears in the LAB\_RESULTS table and as was selected above (i.e. use the LOINC\_CD first if possible).  **TEMP\_MEDICAL:** Code as it appears in the table |  |
| MEASUREMENT\_SOURCE\_CONCEPT\_ID | **LAB\_RESULTS:**  LOINC\_CD,  PROC\_CD  **TEMP\_MEDICAL:**  DIAG1-DIAG5; PROC1-PROC3, PROC\_CD | **LAB\_RESULTS** use When LOINC was selected section 3.1. with filters:  WHERE SOURCE\_VOCABULARY\_ID IN ('LOINC')  AND TARGET\_VOCABULARY\_ID IN ('LOINC')  When PROC\_CD was selected, use section 3.1 with filters from 6.3.  **TEMP\_MEDICAL (DIAG1-DIAG5):**  Use Section 3.1.1 and filter using Section 6.1  **TEMP\_MEDICAL (PROC1-PROC3, PROC\_CD):**  Use Section 3.1.1 and filter using Section 6.2 | Use the code in Section 3.1.1. |
| UNIT\_SOURCE\_ VALUE | **LAB\_RESULTS**:  RSLT\_UNIT\_NM  **TEMP\_MEDICAL:**  NULL |  |  |
| VALUE\_SOURCE\_ VALUE | **LAB\_RESULTS:**  RSLT\_TXT  **TEMP\_MEDICAL:**  DIAG1-DIAG5; PROC1-PROC3, PROC\_CD |  |  |

## Table Name: SPECIMEN

The SPECIMEN table contains information identifying biological samples from a person.

Optum does not have clear specimen information so this table will only contain one value representing the fact that no specimen information will be captured.

| **Table 20: SPECIMEN** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| SPECIMEN\_ID | 0 |  |  |
| PERSON\_ID | 0 |  |  |
| SPECIMEN\_CONCEPT\_ID | 0 |  |  |
| SPECIMEN\_TYPE\_CONCEPT\_ID | 0 |  |  |
| SPECIMEN\_DATE | - | NULL |  |
| SPECIMEN\_TIME | - | NULL |  |
| QUANTITY | - | NULL |  |
| UNIT\_CONCEPT\_ID | 0 |  |  |
| ANATOMIC\_SITE\_CONCEPT\_ID | 0 |  |  |
| DISEASE\_STATUS\_CONCEPT\_ID | 0 |  |  |
| SPECIMEN\_SOURCE\_ID | 0 |  |  |
| SPECIMEN\_SOURCE\_VALUE | - | NULL |  |
| UNIT\_SOURCE\_VALUE | - | NULL |  |
| ANATOMIC\_SITE\_SOURCE\_VALUE | - | NULL |  |
| DISEASE\_STATUS\_SOURCE\_VALUE | - | NULL |  |

## Table Name: DEVICE\_EXPOSURE

The *DEVICE\_EXPOSURE*table will be populated with records from **TEMP\_MEDICAL,** where the PROC code was mapped to a standard concept with a DOMAIN\_ID of ‘Device’.

| **Table 21: DEVICE\_EXPOSURE** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| DEVICE\_EXPOSURE\_ID | - | System Generated |  |
| PERSON\_ID | **TEMP\_MEDICAL:**  PATID / PAT\_PLANID |  |  |
| DEVICE\_CONCEPT\_ID | **TEMP\_MEDICAL:**  PROC1-PROC3, PROC\_CD, DIAG1-5 | **DIAG1-DIAG5:**  Use Section 3.1.2 and filter with Section 4.1  **PROC1-PROC3, PROC\_CD:**  Use Section 3.1.2 and filter with Section 4.2 |  |
| DEVICE\_EXPOSURE\_START\_DATE | **VISIT\_OCCURRENCE** VISIT\_END\_DATE |  |  |
| DEVICE\_EXPOSURE\_END\_DATE | - |  |  |
| DEVICE\_TYPE\_CONCEPT\_ID | - | If the record is coming from another table, like the *PROCEDURE\_OCCURENCE*, keep the types that would have been assigned in that table. |  |
| UNIQUE\_DEVICE\_ID | - |  |  |
| QUANTITY | 0 |  |  |
| PROVIDER\_ID | NEW\_PROV  NEW\_PROVCAT | Map NEW\_PROV  to PROVIDER\_SOURCE\_VALUE and NEW\_PROVCAT to  SPECIALTY\_SOURCE\_VALUE in Provider table to extract associated Provider ID. |  |
| VISIT\_OCCURRENCE\_ID | **VISIT\_OCCURRENCE** **–**VISIT\_OCCURRENCE\_ID | Refer to logic in building VISIT\_OCCURRENCE table for linking with VISIT\_OCCURRENCE\_ID |  |
| DEVICE\_SOURCE\_VALUE | **TEMP\_MEDICAL:**  PROC1-PROC3, PROC\_CD, DIAG1-5 |  |  |
| DEVICE\_SOURCE\_CONCEPT\_ID | **TEMP\_MEDICAL:**  PROC1-PROC3, PROC\_CD, DIAG1-5 | **TEMP\_MEDICAL (PROC1-3 & PROC\_CD):**  Use Section 3.1.1 and filter with Section 6.2  **TEMP\_MEDICAL (DIAG1-DIAG5):**  Use Section 3.1.1 and filter with Section 6.1 | Use the code in Section 3.1.1 |

## Table Name: NOTE

The NOTE table captures free text information recorder by a provider about a patient.

Optum does not have clear note information so this table will only contain one value representing the fact that no note information will be captured.

| **Table 22: NOTE** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| NOTE\_ID | 0 |  |  |
| PERSON\_ID | 0 |  |  |
| NOTE\_DATE | - | NULL |  |
| NOTE\_TIME | - | NULL |  |
| NOTE\_TYPE\_CONCEPT\_ID | 0 |  |  |
| NOTE\_TEXT | - | NULL |  |
| PROVIDER\_ID | 0 |  |  |
| NOTE\_SOURCE\_VALUE | - | NULL |  |
| VISIT\_OCCURRENCE\_ID | 0 |  |  |

## Table Name: DEVICE\_COST

The *DEVICE\_COST* table captures the cost of a medical device or supply used on a person.

Optum does not have clear device cost information so this table will only contain one value representing the fact that no note information will be captured.

| **Table 23: DEVICE\_COST** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| DEVICE\_COST\_ID | 0 |  |  |
| DEVICE\_EXPOSURE\_ID | 0 |  |  |
| CURRENCY\_CONCEPT\_ID | 0 |  |  |
| PAID\_COPAY | - | NULL |  |
| PAID\_COINSURANCE | - | NULL |  |
| PAID\_TOWARD\_DEDUCTIBLE | - | NULL |  |
| PAID\_BY\_PAYER | - | NULL |  |
| PAID\_BY\_COORDINATION\_BENEFITS | - | NULL |  |
| TOTAL\_OUT\_OF\_POCKET | - | NULL |  |
| TOTAL\_PAID | - | NULL |  |
| PAYER\_PLAN\_PERIOD\_ID | 0 |  |  |

## Table Name: DOSE\_ERA

The *DOSE\_ERA* table captures the span of time when the person is assumed to be exposed to a constant dose of a specific active ingredient.

Optum does not have clear dose information so this table will only contain one value representing the fact that no dose information will be captured.

| **Table 24: DOSE\_ERA** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| DOSE\_ERA\_ID | 0 |  |  |
| PERSON\_ID | 0 |  |  |
| DRUG\_CONCEPT\_ID | 0 |  |  |
| UNIT\_CONCEPT\_ID | 0 |  |  |
| DOSE\_VALUE | - | NULL |  |
| DOSE\_ERA\_START\_DATE | - | NULL |  |
| DOSE\_ERA\_END\_DATE | - | NULL |  |

## Table Name: FACT\_RELATIONSHIP

The *FACT\_RELATIONSHIP* table captures the relationships between facts in any table of the CDM. These can be person relationships, care site relationships, indication relationships, usage relationships, or facts derived from one another

Optum does not have clear relationship information so this table will only contain one value representing the fact that no relationship information will be captured.

| **Table 25: FACT\_RELATIONSHIP** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| DOMAIN\_CONCEPT\_ID\_1 | 0 |  |  |
| FACT\_ID\_1 | 0 |  |  |
| DOMAIN\_CONCEPT\_ID\_2 | 0 |  |  |
| FACT\_ID\_2 | 0 |  |  |
| RELATIONSHIP\_CONCEPT\_ID | 0 |  |  |

## Table Name: CDM\_SOURCE

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

| **Table 26: CDM\_SOURCE** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| CDM\_SOURCE\_NAME | - | SES:  Optum Extended SES  DOD:  Optum Extended DOD |  |
| CDM\_SOURCE\_ABBREVIATION | - | SES  DOD |  |
| CDM\_HOLDER |  | Janssen R&D |  |
| SOURCE\_DESCRIPTION |  | Clinformatics Extended DataMart (CEDM) is a robust, longitudinally-linked, statistically de-identified database consisting of de-identified individual-level data.  The SES portion of CEDM provides Socio-Economic Status for members with both medical and pharmacy coverage. It may not be linked with the other portions of CEDM (Date of Death, Non-Affiliate). The DOD portion of CEDM provides date of death for its members as well as location. |  |
| SOURCE\_DOCUMENTATION\_REFERENCE |  | SES:  <https://jnj.sharepoint.com/sites/PHM-GCSP-RND/RWE/Pages/DataSourceDetails.aspx?DataSourceItem=39>  DOD:  https://jnj.sharepoint.com/sites/PHM-GCSP-RND/RWE/Pages/DataSourceDetails.aspx?DataSourceItem=38 |  |
| CDM\_ETL\_REFERENCE |  | https://github.com/OHDSI/ETL-CDMBuilder/tree/master/man |  |
| SOURCE\_RELEASE\_DATE |  | SELECT VERSION\_DATE  FROM [\_Version] | Get from the source tables. |
| CDM\_RELEASE\_DATE |  | SELECT CONVERT(VARCHAR(10), GETDATE(),102) | Get the date the run completes on. |
| CDM\_VERSION | - | V5.0 |  |
| VOCABULARY\_VERSION | - | SELECT VOCABULARY\_VERSION  FROM vocabulary  WHERE VOCABULARY\_ID = 'None' | Taken from the Vocabulary loaded into the CDM. |

## Table Name: VISIT\_COST

The *VISIT\_COST* table will house information about the total cost of an inpatient visit.

Key Conventions:

* Use **TEMP\_MEDICAL** to build this table. We take any record that with a claim type of ‘IP’.

| **Table 27: VISIT\_COST** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| VISIT\_COST\_ID | - | A system-generated unique identifier for each visit cost record |  |
| VISIT\_OCCURRENCE\_ID | **VISIT\_OCCURRENCE**: VISIT\_OCCURRENCE\_ID | Refer to logic in building VISIT\_OCCURRENCE table for linking with VISIT\_OCCURRENCE\_ID |  |
| CURRENCE\_CONCEPT\_ID | - | This will be ‘44818668’ for all rows since this is a US claims database and paid in US Dollars |  |
| PAID\_COPAY | **TEMP\_MEDICAL:**  COPAY |  |  |
| PAID\_COINSURANCE | **TEMP\_MEDICAL:**  COINS |  |  |
| PAID\_TOWARD\_DEDUCTIBLE | **TEMP\_MEDICAL:**  DEDUCT |  |  |
| PAID\_BY\_PAYER | NULL |  |  |
| PAID\_BY\_COORDINATION\_OF\_BENEFITS | **TEMP\_MEDICAL:**  AMT\_COB\_SAV |  |  |
| TOTAL\_OUT\_OF\_POCKET | **TEMP\_MEDICAL**:  COINS+ DEDUCT |  |  |
| TOTAL\_PAID | **TEMP\_MEDICAL:**  STD\_COST |  |  |
| PAYER\_PLAN\_PERIOD\_ID | - | Lookup associated PAYER\_PLAN\_PERIOD\_ID. Look up by PERSON\_ID and PROCEDURE\_DATE. If there no match, put NULL. | There should be only one plan associated |

# Code Snippets

## Vocabulary Mapping

Use this code to map source codes to concept ids; change the source\_vocabulary\_id and target\_vocabulary\_id as needed.

### Source to Source

WITH CTE\_VOCAB\_MAP AS (

       SELECT c.concept\_code AS SOURCE\_CODE, c.concept\_id AS SOURCE\_CONCEPT\_ID, c.CONCEPT\_NAME AS SOURCE\_CODE\_DESCRIPTION,

                        c.vocabulary\_id AS SOURCE\_VOCABULARY\_ID, c.domain\_id AS SOURCE\_DOMAIN\_ID, c.concept\_class\_id AS SOURCE\_CONCEPT\_CLASS\_ID,

            c.VALID\_START\_DATE AS SOURCE\_VALID\_START\_DATE, c.VALID\_END\_DATE AS SOURCE\_VALID\_END\_DATE, c.invalid\_reason AS SOURCE\_INVALID\_REASON,

            c.concept\_ID as TARGET\_CONCEPT\_ID, c.concept\_name AS TARGET\_CONCEPT\_NAME, c.vocabulary\_id AS TARGET\_VOCABULARY\_ID, c.domain\_id AS TARGET\_DOMAIN\_ID,

                        c.concept\_class\_id AS TARGET\_CONCEPT\_CLASS\_ID, c.INVALID\_REASON AS TARGET\_INVALID\_REASON,

            c.STANDARD\_CONCEPT AS TARGET\_STANDARD\_CONCEPT

       FROM CONCEPT c

       UNION

       SELECT source\_code, SOURCE\_CONCEPT\_ID, SOURCE\_CODE\_DESCRIPTION, source\_vocabulary\_id, c1.domain\_id AS SOURCE\_DOMAIN\_ID, c2.CONCEPT\_CLASS\_ID AS SOURCE\_CONCEPT\_CLASS\_ID,

                                        c1.VALID\_START\_DATE AS SOURCE\_VALID\_START\_DATE, c1.VALID\_END\_DATE AS SOURCE\_VALID\_END\_DATE,stcm.INVALID\_REASON AS SOURCE\_INVALID\_REASON,

                                        target\_concept\_id, c2.CONCEPT\_NAME AS TARGET\_CONCEPT\_NAME, target\_vocabulary\_id, c2.domain\_id AS TARGET\_DOMAIN\_ID, c2.concept\_class\_id AS TARGET\_CONCEPT\_CLASS\_ID,

                     c2.INVALID\_REASON AS TARGET\_INVALID\_REASON, c2.standard\_concept AS TARGET\_STANDARD\_CONCEPT

       FROM source\_to\_concept\_map stcm

              LEFT OUTER JOIN CONCEPT c1

                     ON c1.concept\_id = stcm.source\_concept\_id

              LEFT OUTER JOIN CONCEPT c2

                     ON c2.CONCEPT\_ID = stcm.target\_concept\_id

       WHERE stcm.INVALID\_REASON IS NULL

)

SELECT \*

FROM CTE\_VOCAB\_MAP

/\*EXAMPLE FILTERS\*/

WHERE SOURCE\_VOCABULARY\_ID IN ('ICD9CM')

AND TARGET\_VOCABULARY\_ID IN ('ICD9CM')

### Source to Standard Terminology

WITH CTE\_VOCAB\_MAP AS (

       SELECT c.concept\_code AS SOURCE\_CODE, c.concept\_id AS SOURCE\_CONCEPT\_ID, c.concept\_name AS SOURCE\_CODE\_DESCRIPTION, c.vocabulary\_id AS SOURCE\_VOCABULARY\_ID,

                           c.domain\_id AS SOURCE\_DOMAIN\_ID, c.CONCEPT\_CLASS\_ID AS SOURCE\_CONCEPT\_CLASS\_ID,

                                                   c.VALID\_START\_DATE AS SOURCE\_VALID\_START\_DATE, c.VALID\_END\_DATE AS SOURCE\_VALID\_END\_DATE, c.INVALID\_REASON AS SOURCE\_INVALID\_REASON,

                           c1.concept\_id AS TARGET\_CONCEPT\_ID, c1.concept\_name AS TARGET\_CONCEPT\_NAME, c1.VOCABULARY\_ID AS TARGET\_VOCABUALRY\_ID, c1.domain\_id AS TARGET\_DOMAIN\_ID, c1.concept\_class\_id AS TARGET\_CONCEPT\_CLASS\_ID,

                           c1.INVALID\_REASON AS TARGET\_INVALID\_REASON, c1.standard\_concept AS TARGET\_STANDARD\_CONCEPT

       FROM CONCEPT C

             JOIN CONCEPT\_RELATIONSHIP CR

                        ON C.CONCEPT\_ID = CR.CONCEPT\_ID\_1

                        AND CR.invalid\_reason IS NULL

                        AND cr.relationship\_id = 'Maps To'

              JOIN CONCEPT C1

                        ON CR.CONCEPT\_ID\_2 = C1.CONCEPT\_ID

                        AND C1.INVALID\_REASON IS NULL

       UNION

       SELECT source\_code, SOURCE\_CONCEPT\_ID, SOURCE\_CODE\_DESCRIPTION, source\_vocabulary\_id, c1.domain\_id AS SOURCE\_DOMAIN\_ID, c2.CONCEPT\_CLASS\_ID AS SOURCE\_CONCEPT\_CLASS\_ID,

                                        c1.VALID\_START\_DATE AS SOURCE\_VALID\_START\_DATE, c1.VALID\_END\_DATE AS SOURCE\_VALID\_END\_DATE,

                     stcm.INVALID\_REASON AS SOURCE\_INVALID\_REASON,target\_concept\_id, c2.CONCEPT\_NAME AS TARGET\_CONCEPT\_NAME, target\_vocabulary\_id, c2.domain\_id AS TARGET\_DOMAIN\_ID, c2.concept\_class\_id AS TARGET\_CONCEPT\_CLASS\_ID,

                     c2.INVALID\_REASON AS TARGET\_INVALID\_REASON, c2.standard\_concept AS TARGET\_STANDARD\_CONCEPT

       FROM source\_to\_concept\_map stcm

              LEFT OUTER JOIN CONCEPT c1

                     ON c1.concept\_id = stcm.source\_concept\_id

              LEFT OUTER JOIN CONCEPT c2

                     ON c2.CONCEPT\_ID = stcm.target\_concept\_id

       WHERE stcm.INVALID\_REASON IS NULL

)

SELECT \*

FROM CTE\_VOCAB\_MAP

/\*EXAMPLE FILTERS\*/

WHERE SOURCE\_VOCABULARY\_ID IN ('NDC')

AND TARGET\_VOCABUALRY\_ID IN ('RxNORM')

### Source to Maps to Value

WITH CTE\_VOCAB\_MAP AS (

SELECT c.concept\_code AS SOURCE\_CODE, c.concept\_id AS SOURCE\_CONCEPT\_ID, c.concept\_name AS SOURCE\_CODE\_DESCRIPTION, c.vocabulary\_id AS SOURCE\_VOCABULARY\_ID,

c.domain\_id AS SOURCE\_DOMAIN\_ID, c.CONCEPT\_CLASS\_ID AS SOURCE\_CONCEPT\_CLASS\_ID,

c.VALID\_START\_DATE AS SOURCE\_VALID\_START\_DATE, c.VALID\_END\_DATE AS SOURCE\_VALID\_END\_DATE, c.INVALID\_REASON AS SOURCE\_INVALID\_REASON,

c1.concept\_id AS TARGET\_CONCEPT\_ID, c1.concept\_name AS TARGET\_CONCEPT\_NAME, c1.VOCABULARY\_ID AS TARGET\_VOCABUALRY\_ID, c1.domain\_id AS TARGET\_DOMAIN\_ID, c1.concept\_class\_id AS TARGET\_CONCEPT\_CLASS\_ID,

c1.INVALID\_REASON AS TARGET\_INVALID\_REASON, c1.standard\_concept AS TARGET\_STANDARD\_CONCEPT

FROM CONCEPT C

JOIN CONCEPT\_RELATIONSHIP CR

ON C.CONCEPT\_ID = CR.CONCEPT\_ID\_1

AND CR.invalid\_reason IS NULL

AND cr.relationship\_id = 'Maps To Value'

JOIN CONCEPT C1

ON CR.CONCEPT\_ID\_2 = C1.CONCEPT\_ID

AND C1.INVALID\_REASON IS NULL

)

SELECT \*

FROM CTE\_VOCAB\_MAP

/\*EXAMPLE FILTERS\*/

WHERE SOURCE\_CODE = 'V87.43'

# Concept Id Mapping Filters

## From Medical Claims’ Diagnosis Fields

**DIAG1-5:**

WHERE

AND TARGET\_STANDARD\_CONCEPT ='S'

AND TARGET\_INVALID\_REASON IS NULL

AND SOURCE\_VOCABULARY\_ID IN (

CASE

/\*ICD9-CM\*/

WHEN ICD\_FLAG = 9

THEN 'ICD9CM'

/\*ICD10-CM\*/

WHEN ICD\_FLAG = 10

THEN ('ICD10CM')

END

)

## From Medical Claims’ Procedure Fields

**PROC\_CD,PROC1-3:**

If ICD\_FLAG = 9

Filter when procedure code comes from PROC field:

WHERE SOURCE\_VOCABULARY\_ID IN ('ICD9Proc', 'HCPCS','CPT4')

AND TARGET\_STANDARD\_CONCEPT ='S'

AND TARGET\_INVALID\_REASON IS NULL

AND TARGET\_CONCEPT\_CLASS\_ID NOT IN ('HCPCS Modifier','CPT4 Modifier')

ELSE IF ICD\_FLAG = 10

Filter when procedure code comes from PROC field:

WHERE SOURCE\_VOCABULARY\_ID IN ('ICD10PCS', 'HCPCS','CPT4')

AND TARGET\_STANDARD\_CONCEPT ='S'

AND TARGET\_INVALID\_REASON IS NULL

AND TARGET\_CONCEPT\_CLASS\_ID NOT IN ('HCPCS Modifier','CPT4 Modifier')

## From Other Procedure Fields

WHERE SOURCE\_VOCABULARY\_ID IN ('HCPCS','CPT4')

AND TARGET\_STANDARD\_CONCEPT ='S'

AND TARGET\_INVALID\_REASON IS NULL

AND TARGET\_CONCEPT\_CLASS\_ID NOT IN ('HCPCS Modifier','CPT4 Modifier')

# Concept Type Id Mapping Filters

## From Medical Claims’ Diagnosis Fields

**DIAG1-5:**

If PLACE\_OF\_SERVICE\_SOURCE\_VALUE =’IP’ then do,

If DIAG1 then 38000200,

If DIAG2 then 38000201,

If DIAG3 then 38000202,

If DIAG4 then 38000203,

If DIAG5 then 38000204,

End;

Else then do,

If DIAG1 then 38000230,

If DIAG2 then 38000231,

If DIAG3 then 38000232,

If DIAG4 then 38000233,

If DIAG5 then 38000234,

End;

## From Medical Claims’ Procedure Fields

If PLACE\_OF\_SERVICE\_SOURCE\_VALUE =’IP’ then do;

If PROC1 then 38000251,

If PROC2 then 38000252,

If PROC3 then 38000253,

If PROC\_CD then 38000254,

End;

Else then do;

If PROC1 then 38000269,

If PROC2 then 38000270,

If PROC3 then 38000271,

If PROC\_CD then 38000272,

End;

# Source Concept Id Mapping Filters

## From Medical Claims’ Diagnosis Fields

**DIAG1-5:**

If ICD\_FLAG=9 use the filter: WHERE SOURCE\_VOCABULARY\_ID IN ('ICD9CM')

AND TARGET\_VOCABULARY\_ID IN ('ICD9CM')

If ICD\_FLAG=10 use the filter: WHERE SOURCE\_VOCABULARY\_ID IN ('ICD10','ICD10CM')

AND TARGET\_VOCABULARY\_ID IN ('ICD10CM')

## From Medical Claims’ Procedure Fields

**PROC\_CD,PROC1-3:**

ICD\_FLAG=9:

WHERE SOURCE\_VOCABULARY\_ID IN ('ICD9Proc','HCPCS','CPT4')

AND TARGET\_VOCABULARY\_ID IN ('ICD9Proc','HCPCS','CPT4')

AND TARGET\_CONCEPT\_CLASS\_ID NOT IN ('HCPCS Modifier','CPT4 Modifier')

ICD\_FLAG=10:

WHERE SOURCE\_VOCABULARY\_ID IN ('ICD10PCS','HCPCS','CPT4')

AND TARGET\_VOCABULARY\_ID IN ('ICD10PCS','HCPCS','CPT4')

AND TARGET\_CONCEPT\_CLASS\_ID NOT IN ('HCPCS Modifier','CPT4 Modifier')

## From Other Procedure Fields

WHERE SOURCE\_VOCABULARY\_ID IN ('HCPCS','CPT4')

AND TARGET\_VOCABULARY\_ID IN ('HCPCS','CPT4')

AND TARGET\_CONCEPT\_CLASS\_ID NOT IN ('HCPCS Modifier','CPT4 Modifier')