Janssen Research & Development, Pharmaceutical Companies   
of Johnson & Johnson   
Common Data Model (CDM v4.0)

ETL Mapping Specification for Optum Clinformatics v5.0

Qianli Ma, Erica Voss, & Chris Knoll

Version 2.0

**Table of Contents**

[1 Introduction 3](#_Toc406142666)

[1.1 Abbreviations 3](#_Toc406142667)

[1.2 Conventions Used in Document 3](#_Toc406142668)

[2 Data Mapping 4](#_Toc406142669)

[2.1 Processing Sequence Map 4](#_Toc406142670)

[2.2 Table Name: LOCATION 6](#_Toc406142671)

[2.3 Table Name: PERSON 7](#_Toc406142672)

[2.4 Table Name: OBSERVATION\_PERIOD 8](#_Toc406142673)

[2.5 Table Name: PAYER\_PLAN\_PERIOD 9](#_Toc406142674)

[2.6 Table Name: PROVIDER 11](#_Toc406142675)

[2.7 Table Name: VISIT\_OCCURRENCE 12](#_Toc406142676)

[2.8 Table Name: CONDITION\_OCCURRENCE 15](#_Toc406142677)

[2.9 Table Name: CONDITION\_ERA 18](#_Toc406142678)

[2.10 Table Name: PROCEDURE\_OCCURRENCE 19](#_Toc406142679)

[2.11 Table Name: PROCEDURE\_COST 22](#_Toc406142680)

[2.12 Table Name: DEATH 25](#_Toc406142681)

[2.13 Table Name: DRUG\_EXPOSURE 27](#_Toc406142682)

[2.14 Table Name: DRUG\_COST 31](#_Toc406142683)

[2.15 Table Name: DRUG\_ERA 32](#_Toc406142684)

[2.16 Table Name: OBSERVATION 34](#_Toc406142685)

[2.17 Table Name: ORGANIZATION 37](#_Toc406142686)

[2.18 Table Name: CARE\_SITE 37](#_Toc406142687)

[2.19 Table Name: COHORT 38](#_Toc406142688)

# Introduction

The purpose of this document is to describe the Extract, Transform, Load (ETL) mapping of the licensed data from Optum Clinformatics (United Health Group) into the Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM) V4. This document is specific to Optum Clinformatics v5.0 or later.

## 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 |

## 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 |

# 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 Optum MEDICAL 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.

## Processing Sequence Map

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

### Step I

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

### 

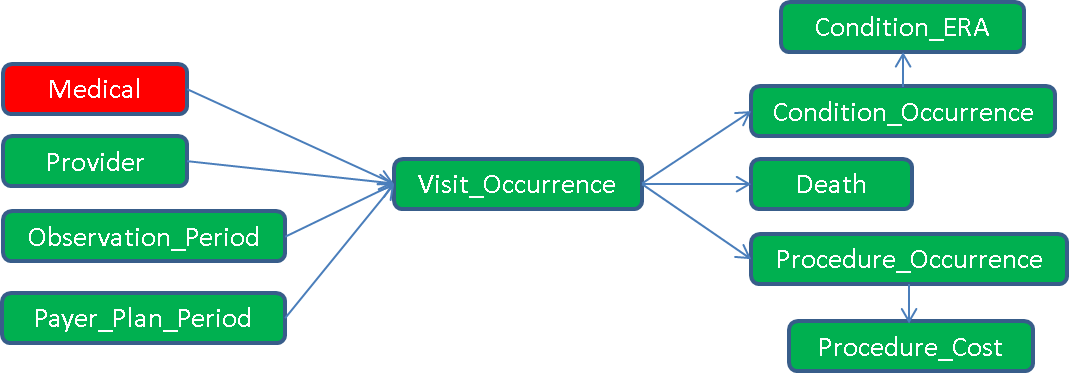
### Step II

Use Optum RX and MEDICAL tables to build Provider table:

### 

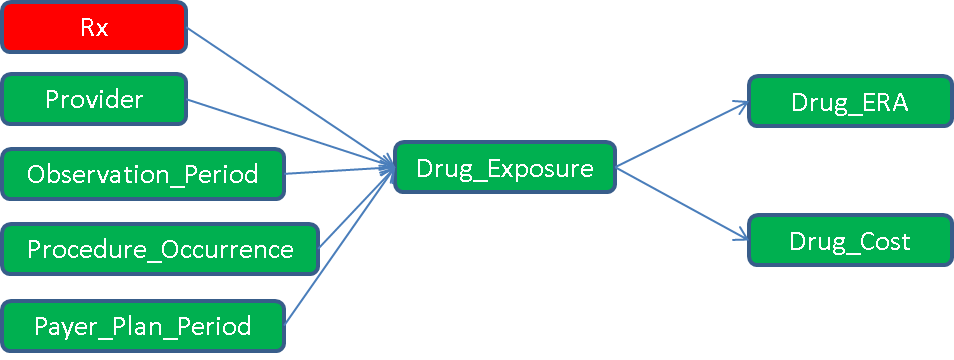
### Step III

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



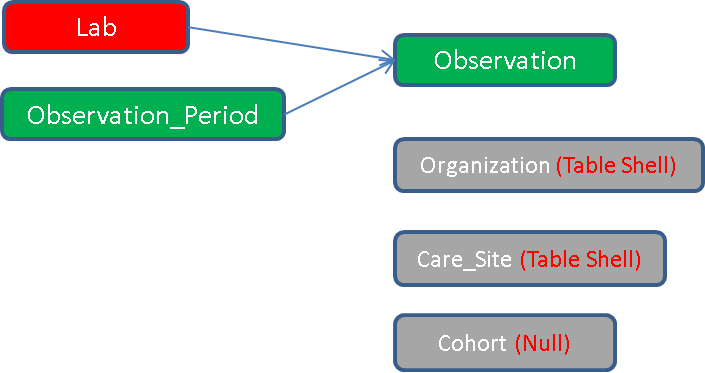
### Step IV

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



### Step V

Use OBSERVATION\_PERIOD and Optum LAB 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):



## Table Name: LOCATION

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

Key conventions:

* Store State in MEMBER\_DETAIL table as LOCATION\_SOURCE\_VALUE.
* Recode invalid codes in State as 'UN':

If LOCATION\_SOURCE\_VALUE NOT IN

('AK','AL','AR','AZ','CA','CO','CT','DC','DE','FL',

'GA','HI','IA','ID','IL','IN','KS','KY','LA','MA',

'MD','ME','MI','MN','MO','MS','MT','NC','ND','NE',

'NH','NJ','NM','NV','NY','OH','OK','OR','PA','PR',

'RI','SC','SD','TN','TX','UT','VA','VT','WA','WI',

'WV','WY') then State ='UN';

Else State = LOCATION\_SOURCE\_VALUE;

* Remove duplicate records before assigning LOCATION\_ID.

| **Table 1: LOCATION** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| LOCATION\_ID | - | System generated. |  |
| ADDRESS\_1 | - | NULL |  |
| ADDRESS\_2 | - | NULL |  |
| CITY | - | NULL |  |
| STATE | *LOCATION\_SOURCE\_VALUE* | Recode source values of State using the code above. |  |
| ZIP | - | NULL |  |
| COUNTY | - | NULL |  |
| LOCATION\_SOURCE\_VALUE | **MEMBER\_ \_DETAIL:**  STATE |  |  |

## Table Name: PERSON

PERSON demographics are sourced from the Optum 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(DOBYR) > min(DOBYR) +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

| **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  U - 8551 | These CONCEPT\_IDs fall under VOCABULARY\_ID = 12 in CONCEPT table. |
| YEAR\_OF\_BIRTH | **MEMBER\_DETAIL:** YRDOB |  |  |
| MONTH\_OF\_BIRTH | - | NULL |  |
| DAY\_OF\_BIRTH | - | NULL |  |
| RACE\_CONCEPT\_ID | - | 0 |  |
| ETHNICITY\_CONCEPT\_ID | - | 0 |  |
| LOCATION\_ID | **MEMBER\_DETAIL**  STATE | 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\_VALUE | **MEMBER\_DETAIL**  GDR\_CD |  |  |
| RACE\_SOURCE\_VALUE | - | NULL |  |
| ETHNICITY\_SOURCE\_VALUE | - | NULL |  |

## Table Name: OBSERVATION\_PERIOD

Optum’s 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 Optum data (10/01/2005) . Note: Depending on your organiation’s cut of data will determine what start bounds you need to enforce. :

CASE

WHEN eligeff < '10/01/2005' THEN '10/01/2005'

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. |  |

## 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 (10/01/2005). Note: Depending on your organiation’s cut of data will determine what start bounds you need to enforce.

CASE

WHEN eligeff < '10/01/2005' THEN '10/01/2005'

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:**  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 and MEDICAL 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 Optum RX table.
* PROV is the provider ID in Optum MEDICAL table but can’t be linked to DEA or NPI, so we will keep it as PROVIDER\_SOURCE\_VALUE too.
* Provider Specialty (PROVCAT) is available in Optum MEDICAL table. We added mapping of Optum provider specialty to OMOP concept - VOCABULARY\_ID = 300.
* Remove duplicate records before assigning PROVIDER\_ID.

| **Table 5: PROVIDER** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| PROVIDER\_ID |  | System generated. |  |
| NPI | - | NULL |  |
| DEA | - | NULL |  |
| SPECIALTY\_CONCEPT\_ID | *SPECIALTY\_SOURCE\_VALUE* | **RX**   **–** set as 38004514 (Unknown Physician Specialty)  **MEDICAL –** PROVCAT: Map it to OMOP VOCABULARY\_ID = 300 | These CONCEPT\_IDs fall under VOCABULARY\_ID = 300. 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 can’t be mapped. |
| CARE\_SITE\_ID | - | 0 |  |
| PROVIDER\_SOURCE\_VALUE | **RX:**  NPI,  DEA  **MEDICAL:**  PROV |  | 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:** NULL  **MEDICAL:**  PROVCAT |  |  |

## 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:

* When pulling from the MEDICAL table, there is a conversion from PAT\_PLANID to PATID. We connect to MEMBER\_DETAILS to get from PAT\_PLANID to PATID.
* **Extract records within OBSERVATION\_PERIODs where a person has both prescription benefits and medical benefits from Optum MEDICAL 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 table pull can be used throughout the loading process. Pulling the following columns will help you throughout the build:

|  |  |
| --- | --- |
| * RVNU\_CD * POS * DIAG1 – DIAG5 * PROC1 – PROC3 * DRG * PROC\_CD * COINS | * COPAY * DEDUCT * STD\_COST * DSTATUS * UNITS * NDC * PAT\_PLANID |

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 PROV and 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 PROV and 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 | **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 | **MEDICAL:** min(FST\_DT) |  |  |
| VISIT\_END\_DATE | **MEDICAL:** max(LST\_DT) |  |  |
| PLACE\_OF\_SERVICE\_CONCEPT\_ID | *PLACE\_OF\_SERVICE\_SOURCE\_VALUE* | Map PLACE\_OF\_SERVICE\_SOURCE\_VALUE to their associated CONCEPT\_IDs:  IP - 9201  OP - 9202  ER - 9203  LTC - 42898160 | These CONCEPT\_IDs fall under VOCABULARY\_ID = 24 in CONCEPT table. |
| CARE\_SITE\_ID | - | NULL |  |
| PLACE\_OF\_SERVICE\_SOURCE\_VALUE | **MEDICAL:** POS RVNU\_CD PROC\_CD | Use the steps mentioned above to create this field. |  |

## 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). Only keep records with valid ICD9 diagnosis (this means source records that do not look like valid codes based on their format will not be brought over):

1. Starts with 0-9, V or E, and non-numeric character is not allowed in other positions.
2. If starts with 0-9 or V, length should be between 3 and 5 (without a decimal); if starts with E, length should be between 4 and 5 (without a decimal).

* In the future there will be a flag (ICD\_FLAG) to say if a claim is using ICD9 or ICD10. However the data is not expected to flow across until 2015. We are not handling for this flag right now and assuming all data is coming across as ICD9.
* Optum removes decimal points from ICD diagnosis 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. Also some ICD procedure codes are misclassified as ICD condition codes in OMOP SOURCE\_TO\_CONCEPT\_MAP, so the following logic should be applied first before removing decimal point in OMOP Vocabulary so you only get ICD conditions:

CASE

WHEN LEN(SOURCE\_CODE) = 3

AND SOURCE\_CODE NOT LIKE '%.%'

THEN 1

WHEN LEN(SOURCE\_CODE) BETWEEN 5 AND 6

AND SUBSTRING(SOURCE\_CODE,4,1) IN ('.')

THEN 1

WHEN SUBSTRING(SOURCE\_CODE, 1, 1) IN ('E') THEN

CASE

WHEN LEN(SOURCE\_CODE) = 4

AND SOURCE\_CODE NOT LIKE '%.%'

THEN 1

WHEN LEN(SOURCE\_CODE) BETWEEN 6 AND 7   
 AND SUBSTRING(SOURCE\_CODE,5,1) IN ('.')

THEN 1

END

ELSE 0  
END

* Always assign its associated VISIT\_START\_DATE as CONDITION\_START\_DATE, and use NEW\_PROV and NEW\_PROVCAT (refer to VISIT\_OCCURRENCE section) extract its ASSOCIATED\_PROVIDER\_ID from 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 | **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 | **MEDICAL:** DIAG1, DIAG2,  DIAG3,  DIAG4,  DIAG5 | Some ICD procedure codes are misclassified as ICD condition codes in OMOP SOURCE\_TO\_CONCEPT\_MAP , so use the logic mentioned above to exclude them, then apply the code below to map source codes:  SELECT DISTINCT REPLACE(SOURCE\_CODE, '.', '') AS SOURCE\_CODE, TARGET\_CONCEPT\_ID  FROM SOURCE\_TO\_CONCEPT\_MAP  WHERE TARGET\_VOCABULARY\_ID = 1 /\*SNOMED\*/  AND SOURCE\_VOCABULARY\_ID = 2 /\*ICD9\*/  AND PRIMARY\_MAP = 'Y'  AND stcm.INVALID\_REASON IS NULL  AND GETDATE() BETWEEN stcm.VALID\_START\_DATE and stcm.VALID\_END\_DATE |  |
| CONDITION\_START\_DATE | **VISIT\_OCCURRENCE:** VISIT\_START\_DATE |  |  |
| CONDITION\_END\_DATE | **-** | NULL |  |
| CONDITION\_TYPE\_CONCEPT\_ID | **VISIT\_OCCURRENCE:** PLACE\_OF\_SERVICE\_SOURCE\_VALUE | If PLACE\_OF\_SERVICE\_SOURCE\_VALUE =’IP’ then do,  If DIAG1 then 38000199,  If DIAG2 then 38000200,  If DIAG3 then 38000201,  If DIAG4 then 38000202,  If DIAG5 then 38000203,  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; | If same ICD code exists in more than one position, keep only the one with low position number.  Map to inpatient/outpatient header to be consistent with PROCEDURE\_OCCURRENCE table.  These CONCEPT\_IDs fall under VOCABULARY\_ID = 37 in CONCEPT table. |
| STOP\_REASON | - | NULL |  |
| ASSOCIATED\_PROVIDER\_ID | **TEMP\_MEDICAL:** NEW\_PROV, NEW\_PROVCAT | Map PROV and 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 | **MEDICAL:**  DIAG1, DIAG2,  DIAG3,  DIAG4,  DIAG5 | Exclude records with invalid ICD9 diagnosis codes. | Use rules mentioned above to exclude invalid ICD9 diagnosis codes. |

## Table Name: CONDITION\_ERA

CONDITION\_ERAs 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\_ERAs 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).

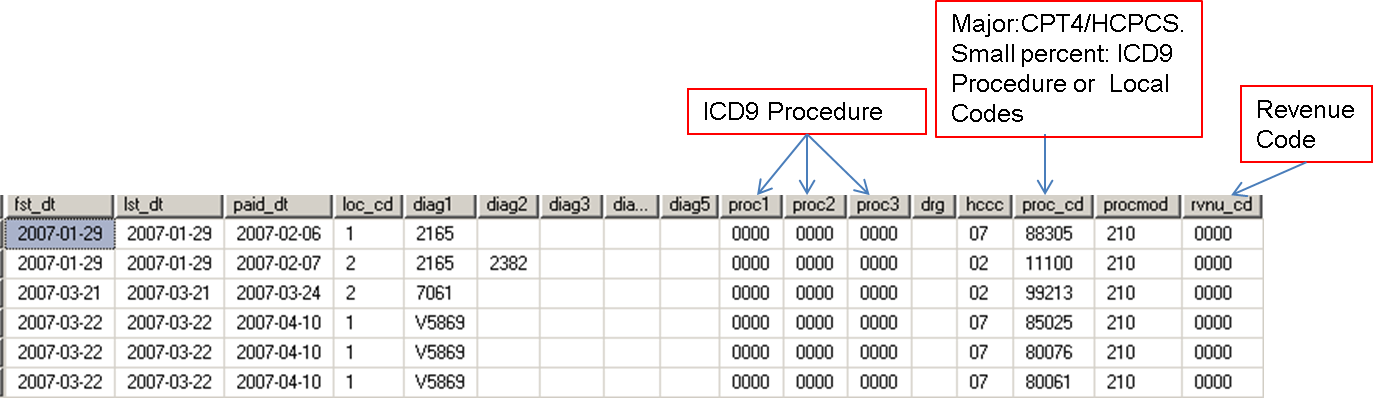
| **Table 8: CONDITION\_ERA** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| CONDITION\_ERA\_ID | - | System generated. |  |
| PERSON\_ID | **CONDITION\_OCCURRENCE:**  PERSON\_ID |  |  |
| CONDITION\_CONCEPT\_ID | **CONDITION\_OCCURRENCE:**  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\_TYPE\_CONCEPT\_ID | - | Apply a 30 day persistence window and label as CONCEPT\_ID 38000247. | CONCEPT\_ID Falls under VOCABULARY\_ID = 37 in CONCEPT table. |
| 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 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 Optum MEDICAL table



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

1. ‘00’, ‘000’, ‘0000’ are usually used as place holder and will be excluded.
2. ICD procedure codes should contains only 2, 3 or 4 digits numeric codes (without their decimals). Exclude codes that are not in this format.
3. 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.
4. Following step 1 through 3, 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.
5. Map PROCEDURE\_SOURCE\_VALUE to **ICD** procedure codes in SOURCE\_TO\_CONCEPT\_MAP.

* 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 PROV and 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 in SOURCE\_TO\_CONCEPT\_MAP.

* 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 | **MEDICAL:** PATID/PAT\_PLANID | At the row level we work with PAT\_PLANID, but PATID is what is written to the CDM. |  |
| PROCEDURE\_CONCEPT\_ID | **MEDICAL:** PROC1,  PROC2,  PROC3  (ICD procedure codes)  **MEDICAL:** PROC\_CD(most are  CPT4/HCPCS code, but some are ICD procedure codes or local codes) | Map source codes to CONCEPT\_ID:  SELECT DISTINCT REPLACE(SOURCE\_CODE, '.', '') AS SOURCE\_CODE, TARGET\_CONCEPT\_ID  FROM SOURCE\_TO\_CONCEPT\_MAP  WHERE SOURCE\_VOCABULARY\_ID in (3,4,5)  AND TARGET\_VOCABULARY\_ID in (3,4,5)  AND PRIMARY\_MAP = 'Y'  AND stcm.INVALID\_REASON IS NULL  AND GETDATE() BETWEEN stcm.VALID\_START\_DATE and stcm.VALID\_END\_DATE | Procedures included in the DRUG\_EXPOSURE table are also included here.  0 will be set for unmapped procedure codes. |
| PROCEDURE\_DATE | **MEDICAL:**  FST\_DT  **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 | 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; | Map to inpatient/outpatient header.  These CONCEPT\_IDs fall under VOCABULARY\_ID = 38 in CONCEPT table. |
| ASSOCIATED\_PROVIDER\_ID | **TEMP\_MEDICAL:** NEW\_PROV,  NEW\_PROVCAT (ICD procedure codes)  PROV,  PROVCAT  (most are  CPT4/HCPCS code, but some are ICD procedure codes or local codes) | Map PROV and 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 |  |  |
| RELEVANT\_CONDITION\_CONCEPT\_ID | **MEDICAL:** DIAG1  Select MIN(DIAG1) for PROC\_CDs but take all DIAG1s that come with PROC1, PROC2, and PROC3. | Some ICD9 procedure codes are misclassified as ICD9 condition codes in OMOP SOURCE\_TO\_CONCEPT\_MAP, so use the logic mentioned in building CONDITION\_OCCURRENCE table to exclude them, then apply the code below to map source codes:  SELECT DISTINCT REPLACE(SOURCE\_CODE, '.', '') AS SOURCE\_CODE,  TARGET\_CONCEPT\_ID  FROM SOURCE\_TO\_CONCEPT\_MAP  WHERE TARGET\_VOCABULARY\_ID = 1 /\*SNOMED\*/  AND SOURCE\_VOCABULARY\_ID = 2 /\*ICD9\*/  AND PRIMARY\_MAP = 'Y'  AND stcm.INVALID\_REASON IS NULL  AND GETDATE() BETWEEN stcm.VALID\_START\_DATE and stcm.VALID\_END\_DATE |  |
| PROCEDURE\_SOURCE\_VALUE | **MEDICAL:** PROC1,  PROC2,  PROC3  **MEDICAL:** PROC\_CD | Clean codes from PROC1, PROC2, PROC3 | See logic above for how to clean ICD procedure codes. |

## 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 a lookup in the STANDARDPRICING table using PAT\_PLANID, CLMID, CLMSEQ, and FST\_DT. It is important to note that some of the lookups for STD\_COST end up getting mapped to multiple rows in the MEDICAL table, this is incorrect and when this happens that STD\_COST should not be used for either record and instead list NULL because we do not know which row to connect it to. Keep all rows in MEDICAL even if they do not map up into STANDARDPRICING, just 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
* Map to DRGs. 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.

LEFT OUTER JOIN Vocabulary.dbo.SOURCE\_TO\_CONCEPT\_MAP stcm

ON stcm.SOURCE\_CODE = tm.drg

AND stcm.TARGET\_VOCABULARY\_ID = 40 /\*DRG\*/

AND stcm.SOURCE\_VOCABULARY\_ID = 40

AND stcm.PRIMARY\_MAP = 'Y'  
 AND po.PROCEDURE\_DATE BETWEEN   
 stcm.VALID\_START\_DATE and stcm.VALID\_END\_DATE

* 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 Optum 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 |  |
| PAID\_COPAY | **MEDICAL:**  COPAY |  |  |
| PAID\_COINSURANCE | **MEDICAL:**  COINS |  |  |
| PAID\_TOWARD\_DEDUCTIBLE | **MEDICAL:**  DEDUCT |  |  |
| PAID\_BY\_PAYER | - | NULL |  |
| PAID\_BY\_COORDINATION\_BENEFITS | **MEDICAL:**  AMT\_COB\_SAV |  |  |
| TOTAL\_OUT\_OF\_POCKET | MEDICAL:  COINS+ DEDUCT |  |  |
| TOTAL\_PAID | **STANDARDPRICING:**  STD\_COST | Lookup using PAT\_PLANID, CLMID, CLMSEQ, and FST\_DT  When a mapping fails to find a STD\_COST in STANDARDPRICING or there were multiple STANDARDPRICING rows per MEDICAL row we will list a NULL instead of STD\_COST. |  |
| DISEASE\_CLASS\_CONCEPT\_ID | **MEDICAL:**  DRG | Map to a DRG that was valid when the procedure was executed.  LEFT OUTER JOIN Vocabulary.dbo.SOURCE\_TO\_CONCEPT\_MAP stcm  ON stcm.SOURCE\_CODE = tm.drg AND stcm.TARGET\_VOCABULARY\_ID = 40 /\*DRG\*/ AND stcm.SOURCE\_VOCABULARY\_ID = 40  AND stcm.PRIMARY\_MAP = 'Y' AND po.PROCEDURE\_DATE BETWEEN stcm.VALID\_START\_DATE and stcm.VALID\_END\_DATE |  |
| REVENUE\_CODE\_CONCEPT\_ID | **MEDICAL:**  RVNU\_CD | Map source codes to CONCEPT\_ID:  Use the following map:  LEFT OUTER JOIN Vocabulary.dbo.SOURCE\_TO\_CONCEPT\_MAP c  ON c.SOURCE\_CODE = tm.rvnu\_cd  AND c.TARGET\_VOCABULARY\_ID = 43  AND c.SOURCE\_VOCABULARY\_ID = 43  AND c.INVALID\_REASON IS NULL  AND c.PRIMARY\_MAP = 'Y'  AND GETDATE() BETWEEN c.VALID\_START\_DATE AND c.VALID\_END\_DATE |  |
| 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. |  |
| DISEASE\_CLASS\_SOURCE\_VALUE | **MEDICAL:** DRG |  |  |
| REVENUE\_CODE\_SOURCE\_VALUE | **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 can be captured by discharge status “Died” or condition in one of the various diagnosis fields containing codes indicating death in MEDICAL table. This table should be loaded when building VISIT\_OCCURRENCE table.

Key conventions:

* Starts from the TEMP\_MEDICAL table generated throughout the VISIT\_OCCURRENCE creation.
* 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. ICD Diagnosis Code

|  |  |  |  |
| --- | --- | --- | --- |
| CODE\_TYPE | SOURCE\_CODE | SOURCE\_CODE\_DESCRIPTION | DEATH\_TYPE\_CONCEPT\_ID |
| ICD9 Code | 7616 | Maternal Death Affecting Fetus or newborn | 38003567 |
| ICD9 Code | 798 | Sudden death, cause unknown | 38003567 |
| ICD9 Code | 7980 | Sudden infant death syndrome | 38003567 |
| ICD9 Code | 7981 | Instantaneous death | 38003567 |
| ICD9 Code | 7982 | Death occurring in less than 24 hours from onset of symptoms, not otherwise explained | 38003567 |
| ICD9 Code | 7989 | Unattended death | 38003567 |
| ICD9 Code | E9131 | Accidental mechanical suffocation by plastic bag | 38003567 |
| ICD9 Code | E978 | Legal execution | 38003567 |

* 1. DRG

This query also provides a way of defining death via DRG. The claim must have the DRG with a VISIT\_END\_DATE between the VALID\_START\_DATE and VALID\_END\_DATE of this lookup.

SELECT 'DRG code indicating death' AS CODE\_TYPE, SOURCE\_CODE, SOURCE\_CODE\_DESCRIPTION, '38003568' AS DEATH\_TYPE\_CONCEPT\_ID, VALID\_START\_DATE, VALID\_END\_DATE

FROM SOURCE\_TO\_CONCEPT\_MAP

WHERE SOURCE\_CODE IN (283, 284, 285, 123, 789, 385)

AND SOURCE\_VOCABULARY\_ID = 40

AND TARGET\_CONCEPT\_ID IN (  
 38000421,38001111,38001112,38001113,38000683,38001500

)

ORDER BY SOURCE\_CODE, INVALID\_REASON

* Only one record will be kept for each patient. The record with latest date of death will be kept. If multiple lines of death records are captured on the same day, the death captured by discharge status has higher priority than those captured by ICD diagnosis codes and the lowest priority goes to records identified by DRG.
* If there are health care visits after 32 days of death date, delete the death record.

| **Table 11: DEATH** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| PERSON\_ID | **MEDICAL:** 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 |  |  |
| DEATH\_TYPE\_CONCEPT\_ID | Discharge status:  **MEDICAL:** DSTATUS  Condition:  **MEDICAL:**  DIAG1, DIAG2, DIAG3,DIAG4,DIAG5  DRG:  **MEDICAL:**  DRG | 38003566 for death captured by Discharge status, 38003567 for death captured by condition, and 38003568 for death captured by DRG | These CONCEPT\_IDs fall under VOCABULARY\_ID = 45 in CONCEPT table. |
| CAUSE\_OF\_DEATH\_CONCEPT\_ID | - | 0 |  |
| CAUSE\_OF\_DEATH\_SOURCE\_VALUE | NULL |  |  |

## Table Name: DRUG\_EXPOSURE

Drug exposures are garnered from RX 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

* **Procedure drug**:
  1. Map PROCEDURE\_SOURCE\_VALUE in PROCEDURE\_OCCURRENCE table to procedure drug and **only keeps those records with mapped CONCEPT\_ID.**

| **Table 12: DRUG\_EXPOSURE** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| DRUG\_EXPOSURE\_ID | - | System generated. |  |
| PERSON\_ID | **RX:** PATID/PAT\_PLANID  **PROCEDURE\_OCCURRENCE:** PERSON\_ID | At the row level we work with PAT\_PLANID, but PATID is what is written to the CDM. |  |
| DRUG\_CONCEPT\_ID | *DRUG\_SOURCE\_VALUE* | Map source codes to CONCEPT\_ID:  **Prescription drug**:  SELECT DISTINCT SOURCE\_CODE, TARGET\_CONCEPT\_ID  FROM SOURCE\_TO\_CONCEPT\_MAP  WHERE  TARGET\_VOCABULARY\_ID = 8 /\*RXNORM\*/  AND SOURCE\_VOCABULARY\_ID = 9 /\*NDC\*/  AND PRIMARY\_MAP = 'Y'  AND INVALID\_REASON IS NULL AND GETDATE() BETWEEN VALID\_START\_DATE and VALID\_END\_DATE  **Procedure drug**: For PROC\_CD, if NDC is NOT NULL then NDC ELSE PROCEDURE\_SOURCE\_VALUE. For all other columns just take the PROCEDURE\_SOURCE\_VALUE  SELECT DISTINCT REPLACE(SOURCE\_CODE, '.', '') AS SOURCE\_CODE,TARGET\_CONCEPT\_ID  FROM SOURCE\_TO\_CONCEPT\_MAP stcm  WHERE  TARGET\_VOCABULARY\_ID = 8 /\*RXNORM\*/  AND  SOURCE\_VOCABULARY\_ID in (3,4,5) /\*ICD procedure/CPT4/HCPCS\*/  AND PRIMARY\_MAP = 'Y'  AND stcm.INVALID\_REASON IS NULL | If unmapped: set 0 for  **Prescription drug**; exclude records for **Procedure drug** |
| DRUG\_EXPOSURE\_START\_DATE | **RX:** FILL\_DT  **PROCEDURE\_OCCURRENCE:** PROCEDURE\_DATE |  |  |
| DRUG\_EXPOSURE\_END\_DATE | - | NULL |  |
| DRUG\_TYPE\_CONCEPT\_ID | **RX:** MAIL\_IND  **PROCEDURE\_OCCURRENCE:**  PROCEDURE\_TYPE\_CONCEPT\_ID | **Prescription drug**:  If MAIL\_IND = ‘Y’ then 38000176 (Prescription dispensed through mail order); else 38000175 (Prescription dispensed in pharmacy)  **Procedure drug**: if  PROCEDURE\_TYPE\_CONCEPT\_ID in (38000251,38000252,38000253,38000254) then 38000180 (Inpatient administration); else 38000179 (Physician administered drug, identified as procedure) | These CONCEPT\_IDs fall under VOCABULARY\_ID = 36 in CONCEPT table. |
| STOP\_REASON | - | NULL |  |
| REFILLS | **RX:** RFL\_NBR | NULL for **Procedure drug** |  |
| QUANTITY | **RX:** sum(QUANTITY)  **PROCEDURE\_OCCURRENCE:**  sum(UNITS) |  |  |
| DAYS\_SUPPLY | **RX :**  sum(DAYS\_SUP) | NULL for **Procedure drug** |  |
| SIG | - | NULL |  |
| PRESCRIBING\_PROVIDER\_ID | **RX:** NPI,  DEA  **PROCEDURE\_OCCURRENCE:**  ASSOCIATED\_PROVIDER\_ID | **Prescription drug**:  Map NPI or DEA to PROVIDER\_SOURCE\_VALUE in PROVIDER table, and extract its associated PROVIDER\_ID;  **Procedure drug**:  ASSOCIATED\_PROVIDER\_ID ID. |  |
| VISIT\_OCCURRENCE\_ID | **PROCEDURE\_OCCURRENCE:**  VISIT\_OCCURRENCE\_ID | **Prescription drug**:  NULL  **Procedure drug**: VISIT\_OCCURRENCE\_ID |  |
| RELEVANT\_CONDITION\_CONCEPT\_ID | **PROCEDURE\_OCCURRENCE:**  RELEVANT\_CONDITION\_CONCEPT\_ID | **Prescription drug**:  NULL  **Procedure drug**: RELEVANT\_CONDITION\_CONCEPT\_ID. |  |
| DRUG\_SOURCE\_VALUE | **RX:** NDC  **PROCEDURE\_OCCURRENCE:**  For PROC\_CD, if NDC is NOT NULL then NDC ELSE PROCEDURE\_SOURCE\_VALUE. For all other columns just take the PROCEDURE\_SOURCE\_VALUE | **Prescription drug**:  Keep all records even if there is no mapping;  **Procedure drug**:  Keep records only when procedure codes are mapped. |  |

## Table Name: DRUG\_COST

DRUG\_CPST information is pulled from the RX table at the same time the DRUG\_EXPOSURE is loaded. Also, it is not appropriate for Optum 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 | Exclude **procedure drug** |
| PAID\_COPAY | **RX:** sum(COPAY) |  |  |
| PAID\_COINSURANCE | - | NULL |  |
| PAID\_TOWARD\_DEDUCTIBLE | **RX:** sum(DEDUCT) |  |  |
| PAID\_BY\_PAYER | - | NULL |  |
| PAID\_BY\_COORDINATION\_BENEFITS | - | NULL |  |
| TOTAL\_OUT\_OF\_POCKET | - | NULL |  |
| TOTAL\_PAID | **RX:** sum(STD\_COST) |  |  |
| INGREDIENT\_COST | - | NULL |  |
| DISPENSING\_FEE | **RX:** sum(DISPFEE) |  |  |
| AVERAGE\_WHOLESALE\_PRICE | **RX:** 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 ca.ANCESTOR\_CONCEPT\_ID /\*ingredient level\*/,

ca.DESCENDANT\_CONCEPT\_ID /\*this is where you set the

DRUG\_EXPOSURE.DRUG\_CONCEPT\_ID to\*/

FROM CONCEPT c1

JOIN CONCEPT\_ANCESTOR ca

ON ca.DESCENDANT\_CONCEPT\_ID = c1.CONCEPT\_ID

JOIN CONCEPT c2

ON c2.CONCEPT\_ID = ca.ANCESTOR\_CONCEPT\_ID

WHERE c1.CONCEPT\_VOCABULARY\_ID = 8

AND c2.CONCEPT\_VOCABULARY\_ID = 8

AND c2.CONCEPT\_LEVEL = 2

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 created 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\_TYPE\_CONCEPT\_ID | - | Apply a 30 day persistence window and label as CONCEPT\_ID 38000182 (Drug era - 30 days persistence window). | This CONCEPT\_ID falls under VOCABULARY\_ID = 36 in CONCEPT table. |
| DRUG\_EXPOSURE\_COUNT | - | Sum up the number of DRUG\_EXPOSUREs for this PERSON\_ID and this CONCEPT\_ID during the exposure window being built. |  |

## Table Name: OBSERVATION

Observations describe symptoms, clinical observations and lab tests etc. about a person are recorded in the OBSERVATION table. Lab observations 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 OMOP VOCABULARY.

Key conventions:

* Only include records with OBSERVATION\_DATE that fall within an OBSERVATION\_PERIODs.
* Units are mapped to UNIT\_CONCEPT\_IDs in the OMOP VOCABULARY (VOCABULARY\_ID = 11 - Unified Code for Units of Measure (UCUM)). However, the OMOP VOCABULARY does not contain all units of measure needed, so VOCABULARY\_ID = 900 – Lab Unit was added. 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, 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 = 6) 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.
* OBSERVATION\_SOURCE\_VALUE: if LOINC\_CD has valid LOINC code, set to LOINC\_CD; else if TST\_DESC has values not in ('', '.','\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'), set to TST\_DESC; else delete records.
* Lab result in Optum Lab 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: **0.000000 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\_AS\_STRING = RSLT\_TXT;

/\*Result as concept code\*/

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

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

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

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

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

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

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

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

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

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

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

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

IF RSLT\_NBR NOT IN (**0.000000**, -**999999.999999**) THEN DO; /\*Result as number\*/

VALUE\_AS\_NUMBER = RSLT\_NBR;

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

END;

ELSE IF VALUE\_AS\_CONCEPT\_ID NE **.** THEN

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

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

| **Table 15:**  **OBSERVATION** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| OBSERVATION\_ID |  | A system generated unique identifier for each observation. |  |
| PERSON\_ID | **LAB:**  PATID/PAT\_PLANID | At the row level we work with PAT\_PLANID, but PATID is what is written to the CDM. |  |
| OBSERVATION\_CONCEPT\_ID | **LAB:**  LOINC\_CD | Map source code to LOINC:  WHERE stcm.SOURCE\_VOCABULARY\_ID= 6  AND stcm.TARGET\_VOCABULARY\_ID =6  AND stcm.PRIMARY\_MAP = 'Y'  AND stcm.INVALID\_REASON IS NULL  AND GETDATE() BETWEEN stcm.VALID\_START\_DATE AND stcm.VALID\_END\_DATE |  |
| OBSERVATION\_DATE | **LAB:**  FST\_DT |  |  |
| OBSERVATION\_TIME | - | NULL |  |
| VALUE\_AS\_NUMBER | **LAB:**  RSLT\_NBR | This is applicable to observations where the result can be converted into numeric value. Set as **NULL** if RSLT\_NBR IN (0.000000, -999999.999999) |  |
| VALUE\_AS\_STRING | **LAB:**  RSLT\_TXT |  |  |
| VALUE\_AS\_CONCEPT\_ID | **LAB:**   ABNL\_CD , RSLT\_TXT | Map result (case insensitive) to the following CONCEPT\_IDs:   |  |  | | --- | --- | | **RESULT** | **CONCEPT\_ID** | | Low | 4267416 | | High | 4328749 | | Normal | 4069590 | | Absent | 4132135 | | Abnormal | 4135493 | | Present | 4181412 | | Final report | 9188 | | Negative | 9189 | | Not found | 9190 | | Positive | 9191 | | Trace | 9192 |   Only applicable for qualitative lab results and set as **0** when not mapped. | These CONCEPT\_IDs fall under VOCABULARY\_ID = 1 and CONCEPT\_CLASS =’Qualifier value’ in CONCEPT table. |
| UNIT\_CONCEPT\_ID | **LAB:** RSLT\_UNIT\_NM | Map source codes (case sensitive and accent sensitive) to UCUM:  WHERE stcm.SOURCE\_VOCABULARY\_ID IN (11, 900)  AND stcm.TARGET\_VOCABULARY\_ID = 11  AND stcm.PRIMARY\_MAP = 'Y'  AND stcm.INVALID\_REASON IS NULL  AND GETDATE() BETWEEN stcm.VALID\_START\_DATE AND stcm.VALID\_END\_DATE  Only applicable for quantitative lab results, set as **0** when not mapped. |  |
| RANGE\_LOW | **LAB:**  LOW\_NRML |  |  |
| RANGE\_HIGH | **LAB:** HI\_NRML |  |  |
| OBSERVATION\_TYPE\_CONCEPT\_ID | **LAB:**  RSLT\_NBR, ABNL\_CD , RSLT\_TXT | 38000277 – Observation numeric result  38000278 – Observation text  38000279 - Lab observation concept code result | These CONCEPT\_IDs fall under  VOCABULARY\_ID=39 in CONCEPT Table |
| ASSOCIATED\_PROVIDER\_ID | - | NULL |  |
| VISIT\_OCCURRENCE\_ID | - | NULL |  |
| RELEVANT\_CONDITION\_CONCEPT\_ID | - | NULL |  |
| OBSERVATION\_SOURCE\_VALUE | **LAB:**  LOINC\_CD | If the LOINC\_CD is not in the right format, take the TST\_DESC |  |
| UNIT\_SOURCE\_VALUE | **LAB:**  RSLT\_UNIT\_NM |  |  |

## Table Name: ORGANIZATION

The ORGANIZATION table contains a list of uniquely identified health care organizations (hospitals, clinics, practices, etc.). This is the highest level of the health care hierarchy. Most organizations contain multiple CARE\_SITEs, but each CARE\_STITE belongs to one Organization. Location information about the organization is stored in the location table and referenced by the foreign key ORGANIZATION\_LOCATION\_ID.

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

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

## 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 17: CARE\_SITE** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| CARE\_SITE\_ID | 0 |  |  |
| LOCATION\_ID | - | NULL |  |
| 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 18: COHORT** | | | |
| --- | --- | --- | --- |
| **Destination Field** | **Source Field** | **Applied Rule** | **Comment** |
| COHORT\_ID |  |  |  |
| COHORT\_CONCEPT\_ID |  |  |  |
| COHORT\_START\_DATE |  |  |  |
| COHORT\_END\_DATE |  |  |  |
| SUBJECT\_ID |  |  |  |
| STOP\_REASON |  |  |  |