OMOP Common Data Model (CDM V5.3)
ETL Mapping Specification Premier

Version 19.0
August 2018

Rupa Makadia & Jamie Weaver

Revision History

Date	Author	Comments
2017/12/19	Jamie Weaver	Update CDM to V5.2
2018/08/21	Jamie Weaver	Update CDM to V5.2

Table of Contents

1.0 Introduction

This document reflects the requirements, assumptions, business rules, and transformations for the implementation of the Common Data Model Version 5.0 (CDM) as implemented by Rupa Makadia, Epidemiology Analytics, Janssen Research and Development.

The purpose of this document is to describe the ETL mapping of the proprietary or licensed data from ***Premier*** into the OMOP Common Data Model. All table references made to the source database, Premier, will be referenced in bold and italic.

Premier is a hospital based system that houses inpatient and outpatient visits from 619 hospitals with over 62 million discharges. Premier is a hospital based system that captures visits and is not a claims database. Each visit is linked to billing records. The data captures 1 in every 5 inpatient stays in the US.

The document is composed of two main sections:

- Source Data Mapping. Describes major tables of the CDM schema and special data handling required for each table.
- Source Independent Data Mapping. Describes mapping process of the drug and condition eras.

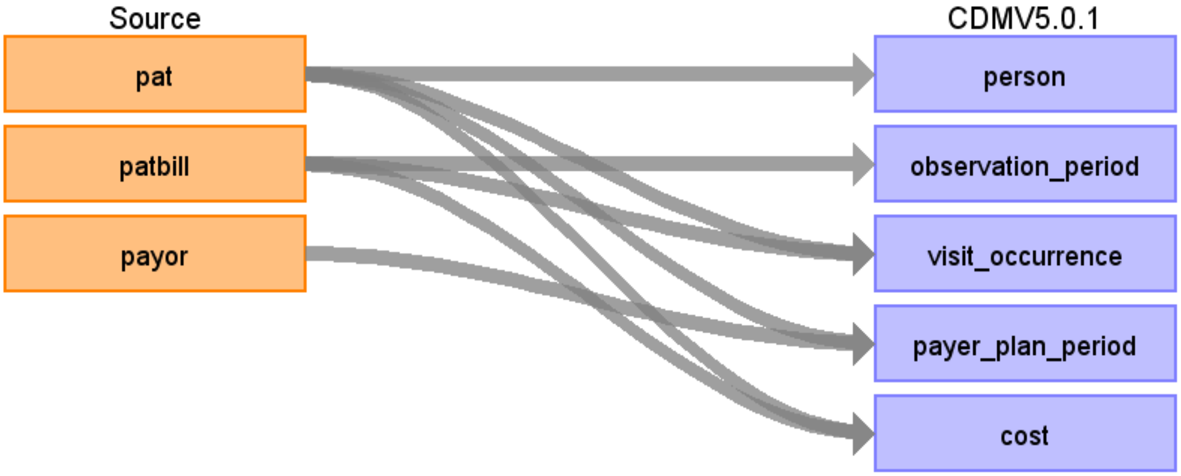
In each section, the tables and their mapping are individually reviewed along with any source-specific rules and exceptions.

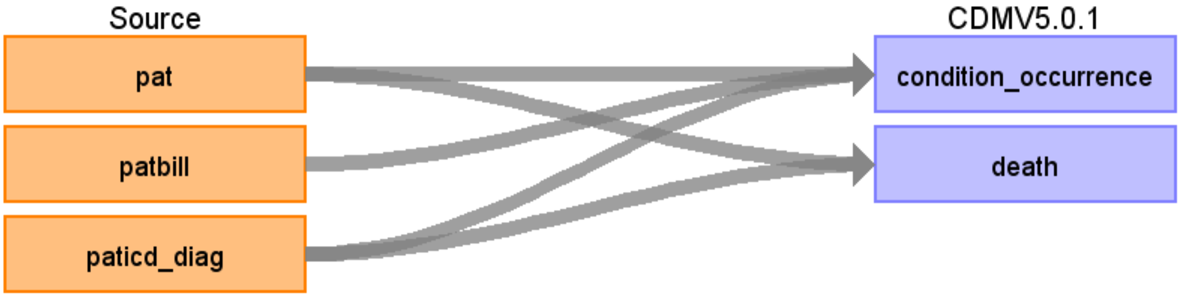
The intended audience for this document is researchers wishing to use the experience and learning in their own CDM construction.

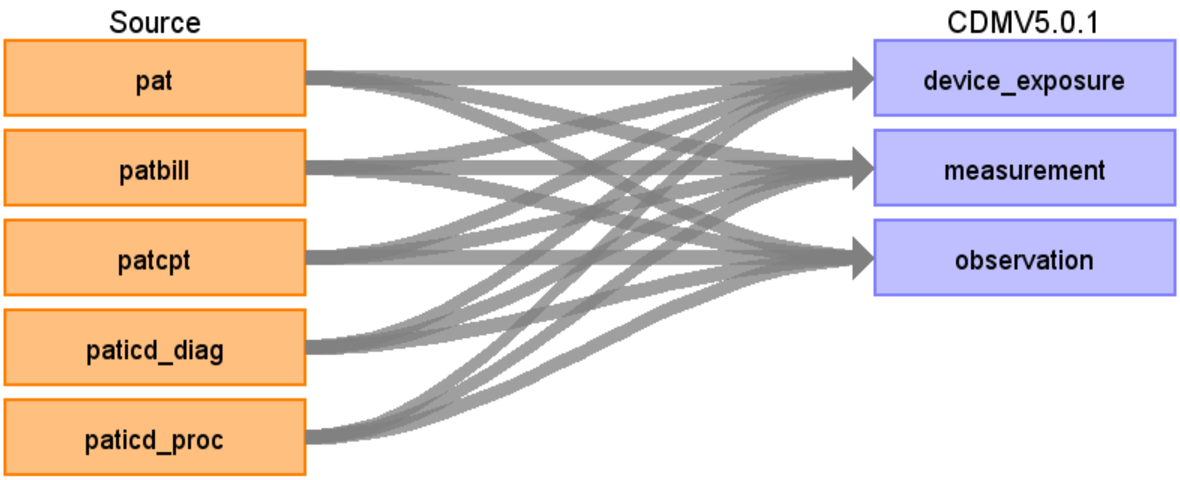
2.0 Source Data Mapping Approach

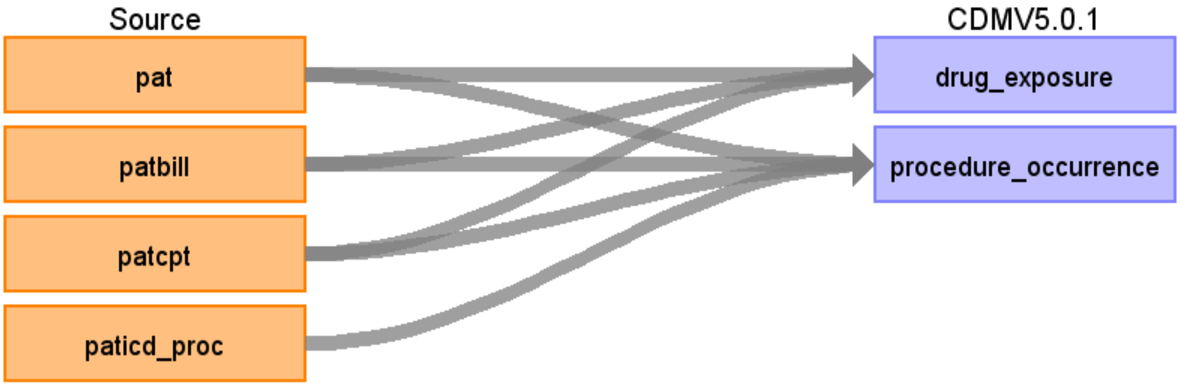
The figures below represent the general approach to mapping the source data tables that comprise Premier to the CDM data schema. Additional ancillary tables and look up tables are included in the chart. The following tables are null: ORGANIZATION and COHORT. The blue boxes represent the input tables in Premier and the green boxes represent the resulting CDM tables. For example, information in the ***PAT*** source table maps to the CDM PERSON table, the VISIT_OCCURRENCE table, the OBSERVATION_PERIOD table, the PAYER_PLAN_PERIOD_TABLE, and the OBSERVATION_TABLE.

In addition to the high-level table mapping that is represented below, a review was conducted of each source data field that is available in all the following tables: ***PAT, PATBILL, PATICD_DIAG, PATICD_PROC*** and ***PATCPT***. The mapping at the field level and justification of why selected fields are not included is in the appendix.



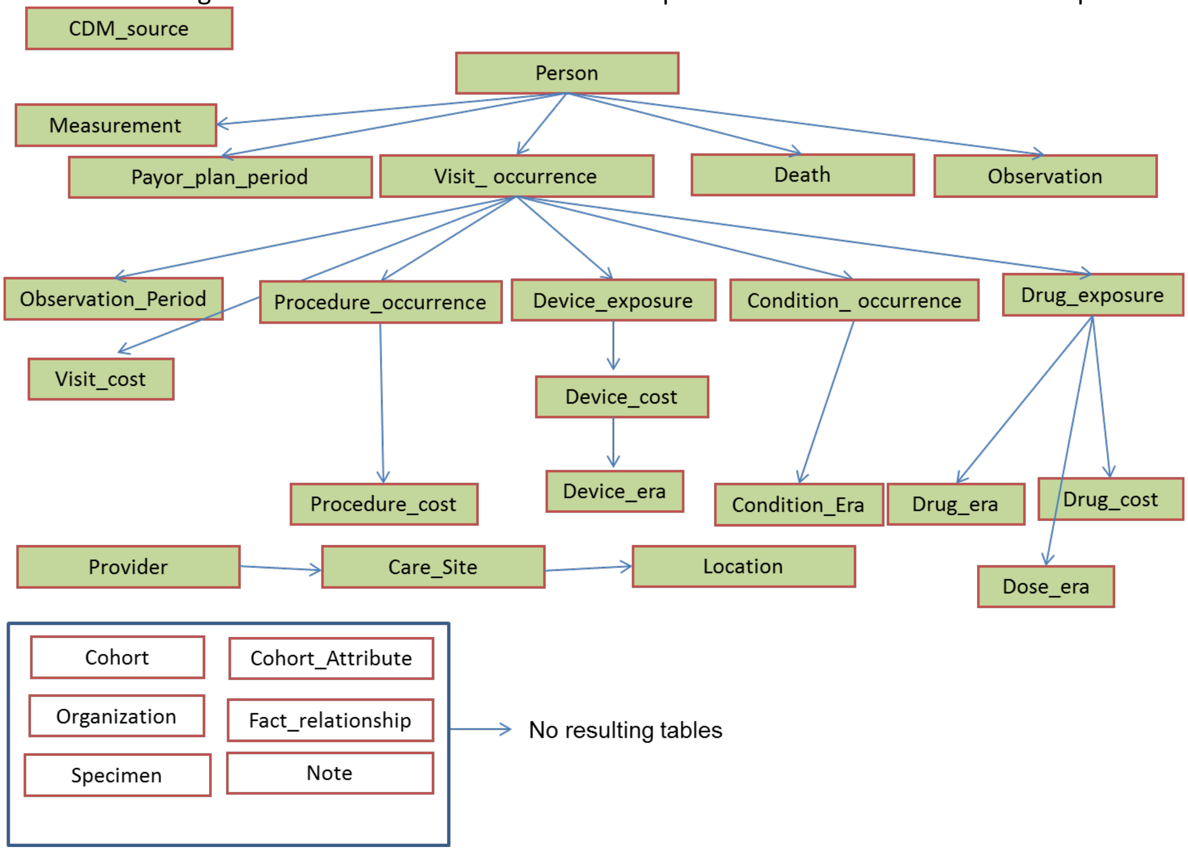



Source
pat
patbill
patcpt
paticd_diag
paticd_proc
CDMV5.0.1
device_exposure
measurement
observation


Source
pat
patbill
patcpt
paticd_proc
CDMV5.0.1
drug_exposure
procedure_occurrence

3.0 Source Data Mapping

The figure below illustrates the order in which the CDM tables are generated. The **VISIT_OCCURRENCE table must be generated first** because procedure occurrence, device exposure, condition occurrence, and drug exposure dates are subsequently generated using visit start date. The start and end date of each visit are derived from the maximum number of service days recorded during a visit. The service days for each visit are in the ***PATBILL*** table where, for each visit, the maximum value in this field is obtained. The logic transformation for these dates is explained in the sections for each respective table.



3.1.1 TABLE NAME: CDM_SOURCE

The CDM_SOURCE table houses metadata about the version of the CDM that is populated, including key elements such as the vocabulary version used in generating the CDM.

Destination Field	Source Field	Applied Rule	Comment
CDM_SOURCE_NAME		Premier	
CDM_SOURCE_ABBRIVIATION		Premier	
CDM_HOLDER		Janssen R&D	
SOURCE_DESCRIPTION		Anonymized hospital transactional database from over 500 hospitals from 2000-present day includes inpatient, outpatient, and emergency room visits. The database is a visit-oriented database, with each visit having its own unique id. Conditions are coded as ICD9 codes and procedures are coded both in ICD9, CPT, and HCPCS procedure codes. Drugs, labs, and other procedures are coded as a standard charge code and occur as a transaction. Cost information is associated to each transaction including charges and quantity of each transaction is recorded in the billing table.	
SOURCE_DOCUMENTATION_REFERENCE		http://hicoe.jnj.com/DataSources/Premier	
CDM_ETL_REFERENCE		http://www.ohdsi.org/web/wiki/doku.php?id=documentation:example_etls	
SOURCE_RELEASE_DATE		`SELECT VERSION_DATE FROM [_Version]`	Get from the raw source tables.
CDM_RELEASE_DATE		`SELECT CONVERT(VARCHAR(10), GETDATE(),102)`	Get the date the run completes.
CDM_VERSION		V5.0	

VOCABULARY_VERSION		`SELECT VOCABULARY_VERSION`
		`FROM VOCABULARY`
		`WHERE VOCABULARY_ID = 'None'`
			Taken from the Vocabulary loaded into the CDM.

3.1.2 TABLE NAME: PERSON

In Premier, the ***PAT*** table contains all demographic, admission, and total cost data for each visit. There are multiple entries per person, thus a single record needs to be created to populate the PERSON table. MONTH_OF_BIRTH and DAY_OF_BIRTH is not available in Premier, because age is the only available field. YEAR_OF_BIRTH is calculated from the first transformed admission date. The admission year minus the age results in the YEAR_OF_BIRTH. Some patients (***MEDREC_KEY***) have some visits (***PAT_KEY***) where ***AGE*** is recorded as 999. This is problematic where ***AGE***=999 at a patient's first visit since year of birth is calculated as year of fist visit (date_part(year, min(VISIT_START_DATE))) minus ***AGE***. The ETL calculates year of birth as year of first visit minus age where age does not equal 999. If all patient age records are 999, then we drop that patient and they do not move to the CDM. Since no address information is available in Premier for each person, LOCATION_ID is null. Primary care providers for each person are not known, thus PROVIDER_ID and CARE_SITE is NULL. Race can vary among records for the same person in the ***PAT*** table, so the most common race value is used for these people. Ethnicity is available in the race field so logic is applied to parse out the ethnicity from each record. Hispanic is the only ethnicity available in Premier so for those with ethnicity recorded as Hispanic, their race is considered 'Other'. For populating the ETHNICITY field, if the race is Hispanic then ETHNICITY is assigned Hispanic otherwise the ethnicity is coded as NULL.

Delete any patients that have invalid birth years < 1900 or > the current year. After birth year has been determined delete any individual that has an OBSERVATION_PERIOD that is >= 2 years prior to the YEAR_OF_BIRTH. Due to data discrepancies in Premier additional logic has been applied to generating gender and age. If a person has YEAR_OF_BIRTH that varies over two years then that person is dropped. Also, if a person has multiple genders recorded or unknown gender then those records are dropped. The exclusion criterion for the PERSON table removes about 1% of the population in the database.

The field mapping is performed as follows:

Destination Field	Source Field	Applied Rule	Comment
PERSON_ID	***PAT.MEDREC_KEY***	Field is a randomly generated identifier that is available in Premier	
GENDER_CONCEPT_ID	***PAT.GENDER***	When ***PAT.GENDER***=M then GENDER_CONCEPT_ID=8507 When ***PAT.GENDER***=F then GENDER_CONCEPT_ID=8532 Delete records with Unknown gender	CONCEPT_ID's are VOCABULARY_ID=Gender

YEAR_OF_BIRTH	***VISIT_OCCURRENCE.VISIT_START_DATE***
	PAT.AGE
		`DATE_PART(YEAR, MIN(VISIT_START_DATE)) - AGE WHERE AGE <> 999`
		`YEAR_OF_BIRTH needs to be > 1900 and <=current year`
		`Drop patients for whom all PAT.AGE records = 999`	

MONTH_OF_BIRTH	-	NULL	Premier only provides age
DAY_OF_BIRTH	-	NULL	Premier only provides age
TIME_OF_BIRTH	-	NULL	Premier only provides age
BIRTH_DATETIME	-	NULL	Premier only provides age

Destination Field	Source Field	Applied Rule	Comment
RACE_CONCEPT_ID	***PAT.RACE***	When ***PAT.RACE***='W' then RACE_CONCEPT_ID=8527 When ***PAT.RACE***='B' then RACE_CONCEPT_ID =8516 When ***PAT.RACE***='H' then RACE_CONCEPT_ID=0 and ETHNICITY_CONCEPT_ID=38003563 Race value of 'O' and 'U' gets mapped to 0	Premier combines both race and ethnicity into one field. Ethnicity is removed from race. If multiple race records per person, see logic to obtain the max value of race that occurs in all records.
ETHNICITY_CONCEPT_ID	***PAT.RACE, PAT.HISPANIC_IND***	When ***PAT.RACE='H' or PAT.HISPANIC_IND***='Y' then ETHNICITY_CONCEPT_ID=38003563 When ***PAT.HISPANIC***='N' then ETHNICITY_CONCEPT_ID=38003564 Ethnicity value of U gets mapped to 0	If race is not Hispanic set ethnicity to 0
LOCATION_ID	-	NULL	
PROVIDER_ID	-	NULL	
CARE_SITE_ID	-	NULL	
PERSON_SOURCE_VALUE	***PAT.MEDREC_KEY***		
GENDER_SOURCE_VALUE	***PAT.GENDER***		
GENDER_SOURCE_CONCEPT_ID	-	NULL	
RACE_SOURCE_VALUE	***PAT.RACE***		
RACE_SOURCE_CONCEPT_ID	-	NULL	
ETHNICITY_SOURCE_VALUE	***PAT.RACE***		
ETHNICITY_SOURCE_CONCEPT_ID	-	NULL	

3.1.3 TABLE NAME: VISIT_OCCURRENCE

Premier data are visit oriented; thus, each visit has its own unique visit identifier. The ***PAT*** table includes admission date and discharge date for each visit. Each visit is stored as a date but the day of the stay is always coded as the first of the month. Because a person can have more than one visit in the same year-month combination, an additional field is included to preserve the order of visits because the day of month is unavailable. The field ***PAT.DISC_MON_SEQ*** is included to preserve the order of visits based on discharge order. The ***LOS*** field on the ***PAT*** table is populated for inpatient stays and is recorded as the number of 24 hour increments that a patient spends in the hospital. The ***LOS*** field is usually off by one calendar day for inpatient visits. Outpatient visits have a ***LOS*** of 0 and are typically only one day in length which is represented in the billing tables. For about 6% of outpatient visits, a single visit is comprised of multiple procedures that occur on different days for procedures such as chemotherapy or dialysis. The Premier billing system does not separate these into individual visits and for purposes of the ETL these will be considered one continuous visit. The ***PATBILL*** table houses billing records that occur each calendar day during the visit. Some visits may include a service day record of zero, which are considered pre-visit tests or procedures. For the purposes of the ETL, the information is considered as the first day of the visit, which can occur for inpatient and outpatient stays. The length of the stay is determined by the ***PAT_BILL*** table using the field ***SERV_DAY.*** The max service day is determined for each person. Logic is defined to take each person, visit, and sequence order and add the maximum number of service days to their stay starting with the first of the month. Each subsequent visit that occurs within the same month as the previous visit will use the previous end date +1 as the visit start date, and then the service days are added to the start date to determine to end date. The same logic is repeated until all visits have a valid start date and end date with no overlaps between subsequent visits. For those visits that have an admission date and discharge date in different months, the logic works backward to obtain the visit start date from the max service days and visit end date. If there are multiple visits that occur in the same admission month that span over months, the records may overlap, which affects roughly 0.10% of patients. For those records that have service days greater than the number of days in the month, the remaining days are truncated to the last day of the month. Logic guarantees that length of visits is preserved as well as the sequential order of visits. The specific day of the month in start date or end date is not necessary accurate, and the time between to two visits is not necessary accurate. The algorithm will always set the new visit start date plus 1 day or use the backward logic and assume that the visit end date is the first day of the new month.

An example for both sets of logic is displayed below:

Raw Data						Transformed Data			
MEDREC_KEY	PAT_KEY	ADM_DATE	DISC_DATE	MAX(SERV_DAY)	DISC_MON_SEQ	PERSON_ID	VISIT_OCCURRENCE_ID	VISIT_START_DATE	VISIT_END_DATE
1120230975	1175536237	10/1/2010	10/1/2010	4	1	1120230975	1175536237	10/1/2010	10/4/2010
1120230975	1228455080	10/1/2010	11/1/2010	10	1	1120230975	1228455080	10/23/2010	11/1/2010
1120230975	1430830346	1/1/2011	1/1/2011	7	1	1120230975	1430830346	1/1/2011	1/7/2010
1120230975	1511534818	2/1/2011	2/1/2011	6	1	1120230975	1511534818	2/1/2011	2/6/2011
1120230975	1560724070	2/1/2011	4/1/2011	48	1	1120230975	1560724070	2/13/2011	4/1/2011
1120230975	1569075673	5/1/2011	5/1/2011	8	1	1120230975	1569075673	5/1/2011	5/8/2011
1120230975	1568584891	5/1/2011	5/1/2011	3	2	1120230975	1568584891	5/9/2011	5/11/2011
1120230975	1606647218	7/1/2011	7/1/2011	3	1	1120230975	1606647218	7/1/2011	7/3/2011
1120230975	1606676096	7/1/2011	7/1/2011	7	2	1120230975	1606676096	7/4/2011	7/10/2011
1120230975	1649550323	8/1/2011	8/1/2011	10	1	1120230975	1649550323	8/1/2011	8/10/2011
1120230975	1649545709	8/1/2011	8/1/2011	11	2	1120230975	1649545709	8/11/2011	8/21/2011
1120230975	1856443013	10/1/2011	10/1/2011	10	1	1120230975	1856443013	10/1/2011	10/10/2011

Logic for ER stays is varied due to data changes from Premier in identifying ER visits. ER Visits in Premier are identified through point of origin or admission source. If a patient visits the ER and then leaves, the visit is considered ER. If an inpatient stay results from an ER visit, that is identified as an ER-to-Inpatient stay. Inpatient stays and outpatient stays with no associated ER visit are simply consider inpatient and outpatient stays, respectively. Additional logic has been added to constrain dates and fields which reflect changes in Premier's classification of emergency room visits. In Premier, the value of admission source that designates emergency room was discontinued 7/1/2010 because it no longer was a required variable for CMS. Point of origin represents the last physical location of a patient before entering the hospital. This field is populated in Premier after 7/1/2010 with emergency room visits. Thus, a combination of point of origin, admission source, and admission type is used to determine if a patient had a valid ER stay for Premier. ***LOS***, the length of stay field, is not used because each billing record corresponds to a service day in the ***PATBILL*** table.

Admitting and discharge information is captured in Premier as the place of service from which the patient arrived and the place to service to which the patient is discharged.

TODO: The ***READMIT*** table includes the field ***READMIT.DAYS_FROM_PRIOR***, which reports the exact number of days between the discharge date of an inpatient stay and the admission date of the subsequent inpatient stay. Build into visit logic to accurately assign number of days between visits. The visit_occurrence logic described above estimates this value. When patients are discharged from inpatient visits twice in the same month, the visit_occurrence logic assumes a single day gap between visits, which underestimates the true number of days between visits by -0.9085 days on average. Conversely, for discharges that happen in difference months, the visit_occurrence logic overestimates the true number of days between visits by 0.9333 days. Across all inpatient visits, the visit_occurrence logic slightly overestimates the true number of days between visits by 0.3098 days. 80% of the visit_occurrence logic estimates are approximately +/- 2 weeks of the true value, and outliers are observed. 20% of the intervals are >2 weeks which is likely unacceptable precision for readmission studies. We will build ***READMIT.DAYS_FROM_PRIOR*** into the ETL in an upcoming sprint. The query for evaluating the correspondence between the visit_occurrence algorithm and ***READMIT.DAYS_FROM_PRIOR*** is below.

The field mapping is performed as follows:

Destination Field	Source Field	Applied Rule	Comment
VISIT_OCCURRENCE_ID	***PAT.PAT_KEY***		
PERSON_ID	***PAT.MEDREC_KEY***		
VISIT_CONCEPT_ID		When ***PAT.ADM_DATE*** <= '6/1/2010' and ***ADM_SOURCE***=7 and ***I_O_IND*** ='O' then concept_id=9203 When ***PAT.ADM_DATE*** <= '6/1/2010' and ***ADM_SOURCE***=7 and ***I_O_IND*** ='I' then concept_id=262 When ***PAT.ADM_DATE*** >= '7/1/2010' and (***POINT_OF_ORIGIN***=7 or ***ADM_SOURCE*** =1) and ***I_O_IND*** ='O' then concept_id=9203 When ***PAT.ADM_DATE*** >= '7/1/2010' and (***POINT_OF_ORIGIN***=7 or ***ADM_SOURCE*** =1) and ***I_O_IND*** ='I' then concept_id=262 When ***I_O_IND*** ='I' then concept_id=9201 When ***I_O_IND*** ='O' then concept_id=9202	Logic is in the following order: Patient enters through the ER but doesn't result in an inpatient stay will be identified as an ER visit (concept_id = 9203). Patients who entered through the ER but get admitted will be identified as an ER-Inpatient visit (concept_id = 262). Remaining patients either inpatient (concept_id=9201) or outpatient (concept_id=9202)
VISIT_START_DATE	***PAT.ADM_DATE*** ***PATBILL.SERV_DAY*** ***PAT.DISC_MON_SEQ***	See logic in code and explanation above	
VISIT_START_DATETIME	-	NULL	
VISIT_END_DATE	***PAT.DISC_DATE*** ***PATBILL.SERV_DAY*** ***PAT.DISC_MON_SEQ***	See logic in code and explanation above	
VISIT_END_DATETIME	-	NULL	
VISIT_TYPE_CONCEPT_ID		44818517- Visit derived from encounter on claim	
PROVIDER_ID	***PAT.ADM_PHY***		
CARE_SITE_ID	***PAT.PROV_ID***		
VISIT_SOURCE_VALUE	***PAT.I_O_IND***		
VISIT_SOURCE_CONCEPT_ID	-	NULL	

Destination Field	Source Field	Applied Rule	Comment
ADMITTING_SOURCE_CONCEPT_ID	***PAT.POINT_OF_ORIGIN***	if POINT_OF_ORIGIN then ADMITTING_SOURCE_CONCEPT_ID 0 then 8976 1 then 8844 2 then 8716 3 then 8844 4 then 8717 45 then 581384 46 then 8650 5 then 8863 6 then 8844 7 then 8870 8 then 8844 9 then 8844 A then 8761 B then 8536 C then 8536 D then 8717 E then 8883 F then 8546	Hardcoding for now. Eventually point_of_origin values from UB-04 vocab will be added to standardized vocabularies and added to STCM.
ADMITTING_SOURCE_VALUE	***PAT. POINT_OF_ORIGIN***	***PAT. POINT_OF_ORIGIN***	
DISCHARGE_TO_CONCEPT_ID	***PAT.DISC_STATUS***	If DISC_STATUS then DISCHARGE_TO_CONCEPT_ID 1 then 8536, 2 then 8844 3 then 8863, 4 then 8863 5 then 8844, 6 then 8536 7 then 8844, 8 then 8536 9 then 8717, 20 then NULL 21 then 8844, 30 then 8844 40 then 8546, 41 then 8546 42 then 8546, 43 then 8966 50 then 8546, 51 then 8546 61 then 8863, 62 then 8920 63 then 8970, 64 then 8676 65 then 8971, 66 then 581379 69 then 8844, 70 then 8844 71 then 8844, 72 then 8717 81 then 8536, 82 then 581379 83 then 8863, 84 then 8827 85 then 8844, 86 then 8536 87 then 8844, 88 then 8966 89 then 8863, 90 then 581379 91 then 581379, 92 then 8676 93 then 8971, 94 then 581379 95 then 8844, 99 then 8844	
DISCHARGE_TO_SOURCE_VALUE	***PAT.DISC_STATUS***	***PAT.DISC_STATUS***	
PRECEDING_VISIT_OCCURRENCE_ID	***VISIT_OCCURRENCE.VISIT_OCCURRENCE_ID***	For a given person, find the visit prior to this one and reference it here	A foreign key to the VISIT_OCCURRENCE table of the visit immediately preceding this visit

SAMPLE CODE: T-SQL logic

```
--sort table by adm_date and dis_mon_seq, make sure to pick up max_serv_day
DECLARE visit_cursor CURSOR FOR
select  pat.pat_key as id,
      pat.medrec_key as person_id,
      pat.adm_date as StartDate,
      pat.disc_date as EndDate,
      pat.disc_mon_seq,
      case when max(patbill.serv_day)=0 then 1 else max(patbill.serv_day)
end as max_serv_day
from pat
      inner join patbill
on pat.pat_key = patbill.pat_key
where pat.medrec_key in {0}
group by pat.patkey, pat.medrec_key, pat.adm_date, pat.disc_date,
pat.dis_mon_seq
order by pat.disc_date asc, pat.dis_mon_seq asc
;

--create temp variables for cursoring

DECLARE @lastenddate date
DECLARE @laststartdate date
DECLARE @lastendmonth date
DECLARE @currentid int
DECLARE @oldpersonid int
DECLARE @currentpersonid int
DECLARE @currentstartdate date
DECLARE @currentenddate date
DECLARE @currentdismonseq int
DECLARE @currentmaxservday int


SET @laststartdate = '1/1/1900'
SET @lastenddate = '1/1/1900'
SET @lastendmonth = '1/1/1900'
SET @oldpersonid=0

OPEN visit_cursor

FETCH NEXT FROM visit_cursor INTO @currentid, @currentpersonid,
@currentstartdate, @currentenddate, @currentdismonseq, @currentmaxservday

WHILE @@FETCH_STATUS = 0

BEGIN
IF @oldpersonid <> @currentpersonid
      BEGIN

SET @laststartdate = '1/1/1900'
SET @lastenddate = '1/1/1900'
SET @lastendmonth = '1/1/1900'
```

```
set @oldpersonid=@currentpersonid

END

IF @currentstartdate = @currententdate

      BEGIN
            IF @lastendmonth < @currentstartdate    --should only fire on
first record or when dis_mon_seq = 1
                  BEGIN

                        SET @currententdate = case when
dateadd(dd,@currentmaxservday,@currentstartdate) <
dateadd(mm,1,@currentstartdate)
                                    then dateadd(dd,@currentmaxservday-
1,@currentstartdate)    --if endate falls in month, use it
                                    else dateadd(dd,-1,
dateadd(mm,1,@currentstartdate))   --otherwise, set to last day of the month
                                    end
                        SET @laststartdate = @currentstartdate
                        SET @lastendmonth = @currentstartdate
                        SET @lastenddate = @currententdate


                  END

            ELSE   --IF @lastendmonth = @currentstartdate
                  BEGIN

                        SET @currentstartdate = dateadd(dd, case when
@lastenddate < dateadd(dd,-1,dateadd(mm,1,@lastendmonth)) then 1 else 0 end ,
@lastenddate)      --increment prior visit by 1 unless you're already at the
end of the month
                        SET @currententdate = case when
dateadd(dd,@currentmaxservday,@lastenddate)  < dateadd(dd,-
1,dateadd(mm,1,@lastendmonth))
                                    then
dateadd(dd,@currentmaxservday,@lastenddate)
                                    else dateadd(dd,-
1,dateadd(mm,1,@lastendmonth)) end
                        --use prior visit + 1 and add the maxsrvdate, unless
either date exceeds the end of the month, or else just use the end of te
month
                        SET @laststartdate = @currentstartdate
                        SET @lastenddate = @currententdate

                  END

      END
ELSE   --startdate <> enddate, which means the visit spans across months
      BEGIN
            IF @lastendmonth = @currentstartdate
                  BEGIN
                        SET @currentstartdate = case when datediff(dd,
dateadd(dd, case when @lastenddate < dateadd(dd,-
1,dateadd(mm,1,@lastendmonth)) then 1 else 0 end , @lastenddate)
,@currententdate) > @currentmaxservday
```

```
                              then dateadd(dd, -1*(@currentmaxservday-1), 
@currententdate)
                              else dateadd(dd, 1, @lastenddate)
                              end

                        SET @lastendmonth = @currententdate

                        SET @currententdate = case when 
dateadd(dd,@currentmaxservday-1,@currentstartdate) < 
DATEADD(mm,1,@currententdate)

                              then dateadd(dd,@currentmaxservday-
1,@currentstartdate)
                              else dateadd(dd,-
1,DATEADD(mm,1,@currententdate))
                              end

                        SET @lastenddate = @currententdate

                  END
            ELSE  --@lastendmonth < @currentstartdate    --if the visit 
spanning months is the first visit in the month, start it on first day of adm 
month and let it go through into the disc month
                  BEGIN
                        SET @lastendmonth = @currententdate
                        SET @currentstartdate =DATEADD(dd,-
1*(@currentmaxservday-1), @currententdate)
                        SET @lastenddate = @currententdate


                  END

      END

INSERT INTO scratch.dbo.VISIT_OCCURRENCE (VISIT_OCCURRENCE_ID, PERSON_ID, 
VISIT_START_DATE, VISIT_END_DATE)
      VALUES (@currentid, @currentpersonid, @currentstartdate, 
@currententdate)


FETCH NEXT FROM visit_cursor INTO @currentid, @currentpersonid, 
@currentstartdate, @currententdate, @currentdismonseq, @currentmaxservday


END

CLOSE visit_cursor
DEALLOCATE visit_cursor
```

Adding mapping logic for PLACE_OF_SERVICE and PROVIDER

```
select
visit_occurance_id,
person_id,
visit_start_date,
visit_end_Date,
case when adm_date <= '06/01/2010' and pat.adm_source='7' and pat.i_o_ind='O'
then 9203
     when adm_date >= '07/01/2010' and (pat.point_of_origin='7' OR
pat.adm_source='1') and pat.i_o_ind='O' then 9203
     when pat.i_o_ind='I' then 9201
     when pat.i_o_ind='O' then 9202
end as place_of_service_concept_id,
PAT.PROV_ID AS care_Site_id,
PAT.i_o_ind as place_of_service_source_value
from visit_occurrence
join pat on visit_occurrence.visit_occurrence_id=pat.pat_key
```

SQL for evaluating the correspondence between the visit_occurrence algorithm and ***READMIT.DAYS_FROM_PRIOR***

```
with
first_visit as
(
  select
    p.medrec_key
    , p.pat_key
    , p.disc_mon_seq
    , vo.visit_start_date
    , vo.visit_end_date
    , r.days_from_prior
    , row_number() over (partition by p.medrec_key order by p.medrec_key,
p.adm_date, p.disc_mon, p.disc_mon_seq) as rn1
  from native.pat p
  left join native.readmit r
    on p.medrec_key = r.medrec_key
    and p.disc_mon = r.disc_mon
    and p.disc_mon_seq = r.disc_mon_seq
  inner join native.patbill pb
    on p.pat_key = pb.pat_key
  inner join cdm.visit_occurrence vo
    on p.pat_key = vo.visit_occurrence_id
  where p.i_o_ind = 'I'
  group by p.medrec_key, p.pat_key, p.adm_date, p.disc_mon, p.disc_mon_seq,
p.i_o_ind, vo.visit_start_date, vo.visit_end_date, r.days_from_prior
),
second_visit as
(
  select
    p.medrec_key
    , p.pat_key
    , p.disc_mon_seq
    , vo.visit_start_date
    , vo.visit_end_date
    , r.days_from_prior
    , row_number() over (partition by p.medrec_key order by p.medrec_key,
p.adm_date, p.disc_mon, p.disc_mon_seq) - 1 as rn2
  from native.pat p
  left join native.readmit r
    on p.medrec_key = r.medrec_key
    and p.disc_mon = r.disc_mon
    and p.disc_mon_seq = r.disc_mon_seq
  inner join native.patbill pb
    on p.pat_key = pb.pat_key
  inner join cdm.visit_occurrence vo
    on p.pat_key = vo.visit_occurrence_id
  where p.i_o_ind = 'I'
  group by p.medrec_key, p.pat_key, p.adm_date, p.disc_mon, p.disc_mon_seq,
p.i_o_ind, vo.visit_start_date, vo.visit_end_date, r.days_from_prior
),
gap_diffs as
(
  select
      f.medrec_key
      , f.pat_key as first_pat_key
      , s.pat_key as second_pat_key
      , f.disc_mon_seq as first_visit_disc_mon_seq
      , s.disc_mon_seq as second_visit_disc_mon_seq
      , f.visit_end_date as first_visit_end_date
      , s.visit_start_date as second_visit_start_date
      , s.days_from_prior as days_from_prior_pmr
      , s.visit_start_date - f.visit_end_date as days_from_prior_alg
      , (s.visit_start_date - f.visit_end_date) - s.days_from_prior as
days_from_prior_diff --vo algorithm value - premier value
  from first_visit f
  inner join second_visit s --drops last visit since no subsequent visit for
calculating time between
    on f.medrec_key = s.medrec_key and f.rn1 = s.rn2
),
overallStats (concept_id, avg_value, stdev_value, min_value, max_value,
total) as
(
  select
    case when first_visit_disc_mon_seq <> second_visit_disc_mon_seq then 1
                     when  first_visit_disc_mon_seq =
second_visit_disc_mon_seq then 2
            end as concept_id
    , avg(1.0000 * days_from_prior_diff) as avg_value
    , stddev(days_from_prior_diff) as stdev_value
    , min(days_from_prior_diff) as min_value
    , max(days_from_prior_diff) as max_value
    , count(*) as total
  from gap_diffs
  group by concept_id
),
Stats (concept_id, stat_value, total, rn) as
(
  select
    case when first_visit_disc_mon_seq <> second_visit_disc_mon_seq then 1
                     when  first_visit_disc_mon_seq =
second_visit_disc_mon_seq then 2
            end as concept_id
            , days_from_prior_diff
            , count(*) as total
            , row_number() over (partition by concept_id order by
days_from_prior_diff) as rn
  from gap_diffs
  group by concept_id, days_from_prior_diff
),
StatsPrior (concept_id, stat_value, total, accumulated) as
(
  select s.concept_id, s.stat_value, s.total, sum(p.total) as accumulated
  from Stats s
  join Stats p on s.concept_id = p.concept_id and p.rn <= s.rn
  group by s.concept_id, s.stat_value, s.total, s.rn
),
overallStatsAll (concept_id, avg_value, stdev_value, min_value, max_value,
total) as
(
select
  3 as concept_id
  , avg(1.0000 * days_from_prior_diff) as avg_value
  , stddev(days_from_prior_diff) as stdev_value
  , min(days_from_prior_diff) as min_value
  , max(days_from_prior_diff) as max_value
  , count(*) as total
  from gap_diffs
  group by concept_id
),
StatsAll (concept_id, stat_value, total, rn) as
(
  select
  3 as concept_id
  , days_from_prior_diff
  , count(*) as total
  , row_number() over (order by days_from_prior_diff) as rn
  from gap_diffs
  group by concept_id, days_from_prior_diff
),
StatsPriorAll (concept_id, stat_value, total, accumulated) as
(
  select s.concept_id, s.stat_value, s.total, sum(p.total) as accumulated
  from StatsAll s
  join StatsAll p on s.concept_id = p.concept_id and p.rn <= s.rn
  group by s.concept_id, s.stat_value, s.total, s.rn
)
select
  o.concept_id
  , case when o.concept_id = 1 then 'discharges in same month'
                     when o.concept_id = 2 then 'discharges in different
month'
            end as discharge_gap
      , o.total as count_value
      , o.avg_value
      , o.stdev_value
      , o.min_value
      , min(case when p.accumulated >= .10 * o.total then stat_value end) as
p10_value
      , min(case when p.accumulated >= .25 * o.total then stat_value end) as
p25_value
      , min(case when p.accumulated >= .50 * o.total then stat_value end) as
median_value
      , min(case when p.accumulated >= .75 * o.total then stat_value end) as
p75_value
      , min(case when p.accumulated >= .90 * o.total then stat_value end) as
p90_value
```

```
      , o.max_value
from StatsPrior p
inner join overallStats o
  on p.concept_id = o.concept_id
group by o.concept_id, o.total, o.min_value, o.max_value, o.avg_value,
o.stdev_value
union
select
  o.concept_id
  , 'all discharges' as discharge_gap
  , o.total as count_value
  , o.avg_value
  , o.stdev_value
  , o.min_value
  , min(case when p.accumulated >= .10 * o.total then stat_value end) as
p10_value
  , min(case when p.accumulated >= .25 * o.total then stat_value end) as
p25_value
  , min(case when p.accumulated >= .50 * o.total then stat_value end) as
median_value
  , min(case when p.accumulated >= .75 * o.total then stat_value end) as
p75_value
  , min(case when p.accumulated >= .90 * o.total then stat_value end) as
p90_value
  , o.max_value
from StatsPriorAll p
inner join overallStatsAll o
  on p.concept_id = o.concept_id
group by o.concept_id, o.total, o.min_value, o.max_value, o.avg_value,
o.stdev_value
```

3.1.4 TABLE NAME: OBSERVATION_PERIOD

Because of the lack of enrollment data in Premier, the observation period for each patient will be defined by unique visits from the VISIT_OCCURRENCE table for each unique patient. Derived admission and discharge dates are created using the number of service days and the sequence of visits as defined by the field ***PAT.DISC_MON_SEQ*** if the visits occurred twice in the same discharge month/year. (See VISIT_OCCURRENCE specification).

All overlapping visits will be collapsed into one observation period. An overlapping visit is defined by a visit that has an admit date that is within 31 days of the previous discharge date. For example, a patient has an admission date of ‘2011-02-01’ and a discharge date of ‘2011-02-05’ and the next visit occurs in ‘2011-02-19’ and a discharge date of ‘2011-03-01’. These records would be collapsed into a single observation period.

The field mapping is as follows:

Destination Field	Source Field	Applied Rule	Comment
OBSERVATION_PERIOD_ID	-	System generated	
PERSON_ID	***PAT.MEDREC_KEY***		
OBSERVATION_PERIOD_START_DATE	***VISIT_OCCURRENCE.VISIT_START_DATE***		
OBSERVATION_PERIOD_END_DATE	***VISIT_OCCURRENCE.VISIT_END_DATE***		
PERIOD_TYPE_CONCEPT_ID	-	44814725= Period inferred by algorithm	

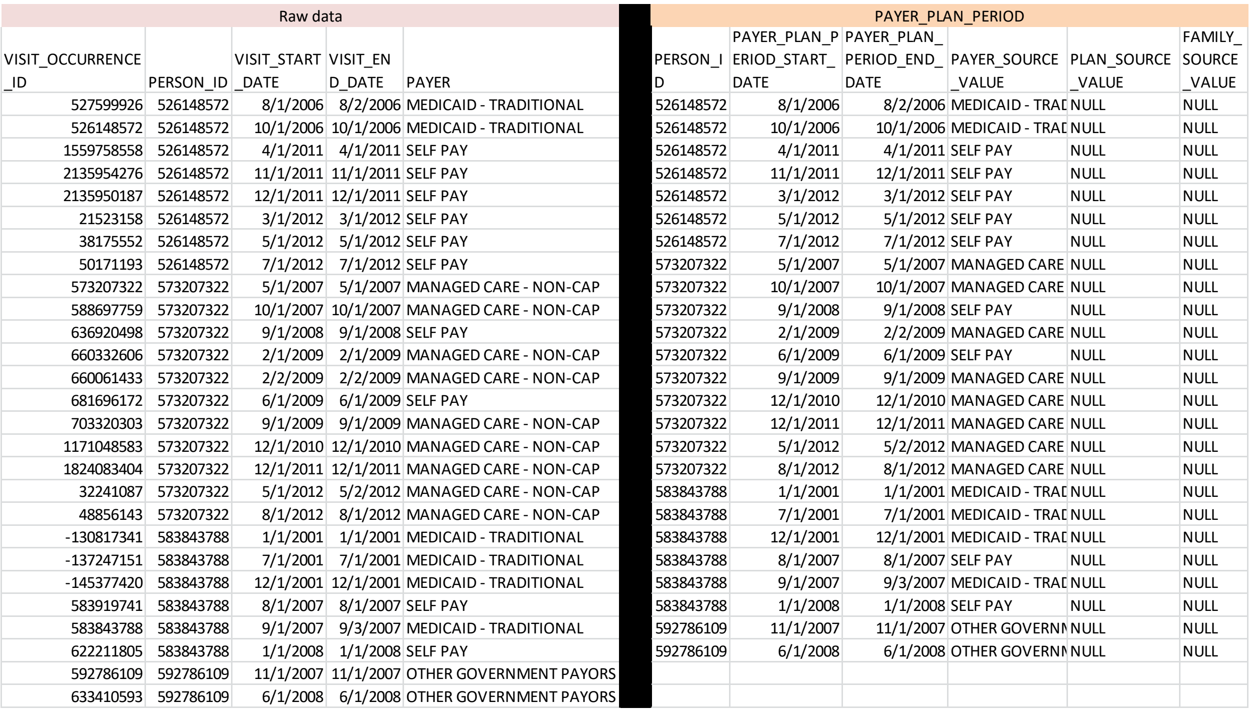
3.1.5 TABLE NAME: PAYER_PLAN_PERIOD

Payer information presented in Premier exists in the ***PAT*** table, in the field ***PAT.STD_PAYOR,*** which represents standard payer categories. Since information about how long the patient remains with a payer or is enrolled is unavailable, the PAYER_PLAN_PERIOD_START_DATE and PAYER_PLAN_PERIOD_END_DATE is the same as the calculated OBSERVATION_PERIOD for each patient. If a patient changes payer within an observation period, then the payer plan period will be segmented to reflect the change in payers. If multiple payers are attributed to one observation period, use visit start and visit end to determine the payer.

The field mapping is as follows:

Destination Field	Source Field	Applied Rule	Comment
PAYER_PLAN_PERIOD_ID	-	System- generated value	
PERSON_ID	***PAT.MEDREC_KEY***		
PAYER_PLAN_PERIOD_START_DATE	***OBSERVATION_PERIOD.OBSERVATION_PERIOD_START_DATE***		
PAYER_PLAN_PERIOD_END_DATE	***OBSERVATION_PERIOD.OBSERVATION_PERIOD_START_DATE***		
PAYER_SOURCE_VALUE	***PAYOR.STD_PAYER_DESC***		
PLAN_SOURCE_VALUE	-	NULL	
FAMILY_SOURCE_VALUE	-	NULL	

An example is represented below:



Raw data						PAYER_PLAN_PERIOD					
VISIT_OCCURRENCE_ID	PERSON_ID	VISIT_START_DATE	VISIT_END_DATE	PAYER		PERSON_ID	PAYER_PLAN_PERIOD_START_DATE	PAYER_PLAN_PERIOD_END_DATE	PAYER_SOURCE_VALUE	PLAN_SOURCE_VALUE	FAMILY_SOURCE_VALUE
527599926	526148572	8/1/2006	8/2/2006	MEDICAID - TRADITIONAL		526148572	8/1/2006	8/2/2006	MEDICAID - TRAD	NULL	NULL
526148572	526148572	10/1/2006	10/1/2006	MEDICAID - TRADITIONAL		526148572	10/1/2006	10/1/2006	MEDICAID - TRAD	NULL	NULL
1559758558	526148572	4/1/2011	4/1/2011	SELF PAY		526148572	4/1/2011	4/1/2011	SELF PAY	NULL	NULL
2135954276	526148572	11/1/2011	11/1/2011	SELF PAY		526148572	11/1/2011	12/1/2011	SELF PAY	NULL	NULL
2135950187	526148572	12/1/2011	12/1/2011	SELF PAY		526148572	3/1/2012	3/1/2012	SELF PAY	NULL	NULL
21523158	526148572	3/1/2012	3/1/2012	SELF PAY		526148572	5/1/2012	5/1/2012	SELF PAY	NULL	NULL
38175552	526148572	5/1/2012	5/1/2012	SELF PAY		526148572	7/1/2012	7/1/2012	SELF PAY	NULL	NULL
50171193	526148572	7/1/2012	7/1/2012	SELF PAY		573207322	5/1/2007	5/1/2007	MANAGED CARE	NULL	NULL
573207322	573207322	5/1/2007	5/1/2007	MANAGED CARE - NON-CAP		573207322	10/1/2007	10/1/2007	MANAGED CARE	NULL	NULL
588697759	573207322	10/1/2007	10/1/2007	MANAGED CARE - NON-CAP		573207322	9/1/2008	9/1/2008	SELF PAY	NULL	NULL
636920498	573207322	9/1/2008	9/1/2008	SELF PAY		573207322	2/1/2009	2/2/2009	MANAGED CARE	NULL	NULL
660332606	573207322	2/1/2009	2/1/2009	MANAGED CARE - NON-CAP		573207322	6/1/2009	6/1/2009	SELF PAY	NULL	NULL
660061433	573207322	2/2/2009	2/2/2009	MANAGED CARE - NON-CAP		573207322	9/1/2009	9/1/2009	MANAGED CARE	NULL	NULL
681696172	573207322	6/1/2009	6/1/2009	SELF PAY		573207322	12/1/2010	12/1/2010	MANAGED CARE	NULL	NULL
703320303	573207322	9/1/2009	9/1/2009	MANAGED CARE - NON-CAP		573207322	12/1/2011	12/1/2011	MANAGED CARE	NULL	NULL
1171048583	573207322	12/1/2010	12/1/2010	MANAGED CARE - NON-CAP		573207322	5/1/2012	5/2/2012	MANAGED CARE	NULL	NULL
1824083404	573207322	12/1/2011	12/1/2011	MANAGED CARE - NON-CAP		573207322	8/1/2012	8/1/2012	MANAGED CARE	NULL	NULL
32241087	573207322	5/1/2012	5/2/2012	MANAGED CARE - NON-CAP		583843788	1/1/2001	1/1/2001	MEDICAID - TRAD	NULL	NULL
48856143	573207322	8/1/2012	8/1/2012	MANAGED CARE - NON-CAP		583843788	7/1/2001	7/1/2001	MEDICAID - TRAD	NULL	NULL
-130817341	583843788	1/1/2001	1/1/2001	MEDICAID - TRADITIONAL		583843788	12/1/2001	12/1/2001	MEDICAID - TRAD	NULL	NULL
-137247151	583843788	7/1/2001	7/1/2001	MEDICAID - TRADITIONAL		583843788	8/1/2007	8/1/2007	SELF PAY	NULL	NULL
-145377420	583843788	12/1/2001	12/1/2001	MEDICAID - TRADITIONAL		583843788	9/1/2007	9/3/2007	MEDICAID - TRAD	NULL	NULL
583919741	583843788	8/1/2007	8/1/2007	SELF PAY		583843788	1/1/2008	1/1/2008	SELF PAY	NULL	NULL
583843788	583843788	9/1/2007	9/3/2007	MEDICAID - TRADITIONAL		592786109	11/1/2007	11/1/2007	OTHER GOVERN	NULL	NULL
622211805	583843788	1/1/2008	1/1/2008	SELF PAY		592786109	6/1/2008	6/1/2008	OTHER GOVERN	NULL	NULL
592786109	592786109	11/1/2007	11/1/2007	OTHER GOVERNMENT PAYORS							
633410593	592786109	6/1/2008	6/1/2008	OTHER GOVERNMENT PAYORS							

3.1.6 TABLE NAME: DEATH

Death is mapped from discharge status and ICD9 codes. The cause of death is not available in Premier. Discharge status from the ***PAT*** table should be used first, and if no codes are found, then ICD9 codes are used. ICD codes that indicate death are found in the source to concept map, see applied rule section below. Keep only one record for each patient, if both discharge status and ICD9 codes indicate death, use discharge status first. No records should be populated for that person 32 days after the death date. The field mapping is as follows:

Destination Field	Source Field	Applied Rule	Comment
PERSON_ID	***PAT.MEDREC_KEY***		
DEATH_DATE	***VISIT_OCCURRENCE .VISIT_END_DATE***		The exact date of death cannot be determined thus the VISIT_END date is used.
DEATH_DATETIME	-	NULL	
DEATH_TYPE_CONCEPT_ID	***PAT.DISC_STATUS*** OR ***PATICD.ICD_CODE***	Logic based on discharge status or ICD9 diagnosis code. If discharge code is present then assign 38003566, Discharge status of PAT.DISC_STATUS in (20, 40, 41, 42) indicates death. Otherwise search PATICD.ICD_CODE records for ICD codes and assign 38003567. To identify death ICD codes, use the following. QUERY: SOURCE TO STANDARD SELECT SOURCE_CODE FROM CTE_VOCAB_MAP WHERE SOURCE_VOCABULARY_ID = 'JNJ_DEATH'	
CAUSE_CONCEPT_ID	-	NULL	
CAUSE_SOURCE_VALUE	-	NULL	
CAUSE_SOURCE_CONCEPT_ID	-	NULL	

3.1.7 TABLE NAME: OBSERVATION

The observation table houses additional demographic and visit data that is housed in Premier. Certain ICD9, CPT, and standard charge codes map to standardized observation table concept. Marital status, admission information, discharge status, and patient type records are specific to Premier and map to non-standard observation table concepts.

PATICD_DIAG.ICD_CODE PATICD_PROC.ICD_CODE, PATCPT.CPT_CODE, and ***PATBILL.STD_CHG_CODE*** map to OBSERVATION.OBSERVATION_CONCEPT_ID using the source to standard cte_vocab_map. These records also map to OBSERVATION.OBSERVATION_SOURCE_CONCEPT_ID using the source to source cte_vocab_map.

PAT.MART_STATUS, PAT.POINT_OF_ORIGIN, PAT.DISC_STATUS, and ***PAT.PATTYPE*** map to set OBSERVATION_CONCEPT_ID codes described in the table below and OBSERVATION_SOURCE_CONCEPT_ID=0.

All inpatient visits (where ***PAT.I_O_IND***=’I’) are associated with a sampling weight ***PAT.PROJ_WGT***. Each inpatient visit record maps to an OBSERVATION table record where OBSERVATION_CONCEPT_ID = 37392832, OBSERVATION_TYPE_CONCEPT_ID = 900000003, and VALUE_AS_NUMBER = PAT.PROJ_WGT. Weights from outpatient visits (***PAT.I_O_IND***=’O’) are all PROJ_WGT=0 and do not move to the OBSERVATION table.

Details on Premier visit sampling weight from
https://www.ncbi.nlm.nih.gov/pubmedhealth/PMH0047457/

"Each hospitalization encounter has an associated statistical weight that allows extrapolation to the volume of hospitalizations estimated for the U.S. as a whole. These weights are based on the inverse of the sampling probabilities associated with each hospital in relationship to the universe of non-federal acute care hospitals, stratified by hospital characteristics, so that the aggregate of hospitalizations approximates the number and distribution of discharges from acute care, non-federal hospitals."

The observation start date is assigned the VISIT_START_DATE. The ASSOCIATED_PROVIDER_ID that is provided is the randomly generated key provided by Premier for the provider that admitted the patient. There are two providers that exist in Premier, the admitting and attending. This ETL makes the decision to use admitting because it is unknown whether the admitting provider, attending provider or another person diagnosed the person.

The field mapping is performed as follows:

Destination Field	Source Field	Applied Rule	Comment
OBSERVATION_ID	-	System-generated	
PERSON_ID	***PAT.MEDREC_KEY***		
OBSERVATION_CONCEPT_ID	***PATCPT.CPT_COD*** ***PATBILL.STD_CHG_CODE*** ***PATICD_PROC.ICD_CODE*** ***PATICD_DIAG.ICD_CODE*** ***PAT.PROJ_WGT***	**For records from PATCPT.CPT_CODE, and PATBILL.STD_CHG_CODE:** QUERY: SOURCE TO STANDARD SELECT TARGET_CONCEPT_ID FROM CTE_VOCAB_MAP WHERE SOURCE_VOCABULARY_ID IN ('CPT4', 'HCPCS', 'JNJ_PMR_OBS_CODE', 'JNJ_PMR_PROC_CHRG_CD') AND TARGET_DOMAIN_ID = 'Observation' **For records from PATICD_PROC.ICD_CODE and PATICD_DIAG.ICD_CODE:** **where ICD_VERSION=9** QUERY: SOURCE TO STANDARD SELECT TARGET_CONCEPT_ID FROM CTE_VOCAB_MAP WHERE SOURCE_VOCABULARY_ID IN ('ICD9CM') AND TARGET_DOMAIN_ID = 'Observation' **For records from PATICD_PROC.ICD_CODE and PATICD_DIAG.ICD_CODE:** **where ICD_VERSION=10** QUERY: SOURCE TO STANDARD SELECT TARGET_CONCEPT_ID FROM CTE_VOCAB_MAP WHERE SOURCE_VOCABULARY_ID IN ('ICD10CM') AND TARGET_DOMAIN_ID = 'Observation' For PAT.MART_STATUS, OBSERVATION_CONCEPT_ID=4053609 For PAT.POINT_OF_ORIGIN, OBSERVATION_CONCEPT_ID=40757183 For PAT.DISC_STATUS, OBSERVATION_CONCEPT_ID= 40757177 For PAT.PATTYPE, OBSERVATION_CONCEPT_ID= 40769091 **For records from PAT.PROJ_WGT:** OBSERVATION_CONCEPT_ID = 37392832	

Destination Field	Source Field	Applied Rule	Comment
OBSERVATION_DATE	***PATBILL.SERV_DAY*** **VISIT_OCCURRENCE.VISIT_START_DATE** ***OR*** ***VISIT_OCCURRENCE.VISIT_START_DATE***	If observation is from ***PATBILL*** use a combination of service day and visit start date unless the service day is greater than the end of the month If observation comes from **PAT.MS_DRG, PAT.PROJ_WGT, PATCPT.CPT_CODE, PATICD_PROC.ICD_CODE, PATICD_DIAG.ICD_CODE** then use visit start date	
OBSERVATION_DATETIME	-	NULL	
OBSERVATION_TYPE_CONCEPT_ID	-	38000281 Observation recorded from EHR with text result If record from PAT.PROJ_WGT, then 900000003 observation numeric result	
VALUE_AS_NUMBER	***PAT.PROJ_WGT***	If I_O_IND='O' then PAT.PROJ_WGT	
VALUE_AS_STRING	***PAT.MART_STATUS*** ***PAT.POINT_OF_ORIGIN*** ***PAT.DISC_STATUS*** ***PAT.PAT_TYPE***	Value_as_string only populated for Premier-specific fields mart_status, point_of_origin, disc_status, and pat_type Marital status values populated directly from PAT.MART_STATUS as 'M', 'S', 'O', or 'U' select point_of_origin_desc from poorgin po join pat p on p.mart_status=po.point_of_origin select disc_status from poorgin po join pat p on p.mart_status=po.point_of_origin select pat_type_desc from pattype p join pat p1 on p1.pat_type=p.pat_type	Use look up values in the text fields.
VALUE_AS_CONCEPT_ID	-	NULL	
QUALIFER_CONCEPT_ID	-	NULL	
UNIT_CONCEPT_ID	-	NULL	
PROVIDER_CONCEPT_ID	***PAT.ADMPHY***		
VISIT_OCCURRENCE_ID	***PAT.PAT_KEY***		

Destination Field	Source Field	Applied Rule	Comment

OBSERVATION_SOURCE_VALUE	***PAT.DRG***
	PATICD.ICD_CODE
	PATCPT.CPT_CODE
	CHARGE CODE
		Standard charge code value:
		`SELECT CONCAT(STD_CHG_DESC, ' / ', HOSP_CHG_DESC) AS SOURCE_VALUE FROM PATBILL A`
		`JOIN CHGMSTR B ON A.STD_CHG_CODE=B.STD_CHG_CODE`
		`JOIN hospchg C ON A.hosp_chg_id=C.hosp_chg_id`	

OBSERVATION_SOURCE_CONCEPT_ID	-	QUERY: SOURCE TO SOURCE
		`SELECT SOURCE_CONCEPT_ID`
		`FROM CTE_VOCAB_MAP`
		`WHERE SOURCE_VOCABULARY_ID IN ('ICD9CM', 'ICD10CM', 'CPT4', 'HCPCS')`
		`AND TARGET_VOCABULARY_ID IN ('ICD9CM', 'ICD10CM', 'CPT4', 'HCPCS')`	

UNITS_SOURCE_VALUE	-	NULL	
QUALIFIER_SOURCE_VALUE	-	NULL	

3.1.8 TABLE NAME: SPECIMEN

Premier does not provide information for specimens collected during a person’s stay.

Destination Field	Source Field	Applied Rule	Comment
SPECIMEN_ID	-	NULL	
PERSON_ID	-	NULL	
SPECIMEN_CONCEPT_ID	-	NULL	
SPECIMEN_TYPE_CONCEPT_ID	-	NULL	
SPECIMEN_DATE	-	NULL	
SPECIMEN_DATETIME	-	NULL	
QUANTITY	-	NULL	
UNIT_CONCEPT_ID	-	NULL	
ANATOMIC_SITE_CONCEPT_ID	-	NULL	
DISEASE_STATUS_CONCEPT_ID	-	NULL	
SPECIMEN_SOURCE_ID	-	NULL	
SPECIMEN_SOURCE_VALUE	-	NULL	
UNIT_SOURCE_VALUE	-	NULL	
ANATOMIC_SITE_SOURCE_VALUE	-	NULL	
DISEASE_STATUS_SOURCE_VALUE	-	NULL	

3.1.9 TABLE NAME: MEASUREMENT

The MEASUREMENT table will house records from ***PATBILL***, ***PATCPT, and PATICD_DIAG*** that have been mapped to the measurement domain. Additionally, procedures that occur on the same day as billing records for operation time will have operation time calculated and recorded in the measurement table.

Measurements are recorded in the ***PATBILL*** table as standard charges. Premier captures the day the measurement is made in the ***SERV_DAY*** field thus, the MEASUREMENT_DATE is determined from the VISIT_START_DATE from VISIT_OCCURRENCE and ***PATBILL.SERV_DAY*** unless the start date is greater than the end of the month, then it's truncated to the end of month. For measurements recorded in the ***PATCPT*** table, the day the measurement was made is unknown so MEASUREMENT_DATE is recorded as VISIT_END_DATE.

In Premier, many procedures are recorded in the ***PATICD_PROC*** table, which includes the day the procedure was performed as ***PATICD_PROC.PROC_DAY***. Certain billing records in ***PATBILL*** include information on surgical operation time. The sample code below the field mapping table returns surgical operation time values in minutes for procedures where operation time billing record(s) happen on the same day. It is assumed that if a procedure and an operating time bill happen on the same day, then the operating time is associated with the procedure. These operation time values move to the MEASUREMENT table and the MEASUREMENT_DATE equals the corresponding PROCEDURE_DATE (which is VISIT_OCCURRENCE + PROC_DAY). To associate a surgical operation time with a procedure: MEASUREMENT.VISIT_OCCURRENCE_ID=PROCEDURE_OCCURRENCE.VISIT_OCCURRENCE_ID AND MEASUREMENT.MEASUREMENT_DATE=PROCEDURE_OCCURRENCE.PROCEDURE_DATE.

There are three providers that exist in Premier, the admitting, attending, and procedure. This ETL makes the decision to use admitting physician for all measurements except operation time because it is unknown whether the admitting provider, attending provider or another person obtained the measurement.

TODO: For operation time measurements, the provider is set as the procedure physician.

Only records that fall within an OBSERVATION_PERIOD are available for each person. The VISIT_OCCURRENCE table must be created before the MEASUREMENT table is created.

The field mapping is performed as follows:

Destination Field	Source Field	Applied Rule	Comment
MEASUREMENT_ID	-	System generated	
PERSON_ID	***PAT.MEDREC_KEY***		
MEASUREMENT_CONCEPT_ID	***PATCPT.CPT_CODE*** ***PATBILL.STD_CHG_CODE*** ***PATICD_DIAG.ICD_CODE*** ***PATBILL.STD_CHG_DESC***	QUERY: SOURCE TO STANDARD SELECT TARGET_CONCEPT_ID FROM CTE_VOCAB_MAP WHERE SOURCE_VOCABULARY_ID IN ('CPT4', 'HCPCS', 'ICD10CM', 'ICD9CM', 'JNJ_PMR_PROC_CHRG	Only capture those records that have a domain map to Measurement.

Destination Field	Source Field	Applied Rule	Comment
		_CD') AND TARGET_DOMAIN_ID = 'Measurement' When operation time measurement values then 3016562	
MEASUREMENT_DATE	**VISIT_OCCURRENCE.VISIT_START_DATE** ***PATBILL.SERV_DAY*** ***Or*** **VISIT_OCCURRENCE.VISIT_END_DATE** ***Or*** **VISIT_OCCURRENCE.VISIT_START_DATE** ***PATICD_PROC.PROC_DAY***		If measurement is from ***PATBILL*** use a combination of service day and visit start date unless the service day is greater than the end of the month If measurement comes from ***PATCPT*** then use visit end date For operation time measurement, a combination of procedure day and visit start date unless the procedure day is greater than the end of the month
MEASUREMENT_DATETIME	-	NULL	
MEASUREMENT_TYPE_CONCEPT_ID	-	For operation time records 45754907-Derived value else 44818701-From physical examination	
OPERATOR_CONCEPT_ID	-	NULL	
VALUE_AS_NUMBER	-	See query below	
VALUE_AS_CONCEPT_ID	-	NULL	
UNIT_CONCEPT_ID	-	For operation time records 8550 Else NULL	
RANGE_LOW	-	NULL	
RANGE_HIGH	-	NULL	
PROVIDER_ID	***PATICD_PROC.PROC_PHY*** ***PAT.ADMPHY***	When operation time ***PATICD_PROC.PROC_PHY*** Else	

Destination Field	Source Field	Applied Rule	Comment
		PAT.ADMPHY	
VISIT_OCCURRENCE_ID	***PAT.PAT_KEY***		

MEASUREMENT_SOURCE_VALUE		`SELECT SOURCE_VALUE FROM`
		`(`
		`SELECT CONCAT(STD_CHG_DESC, ' / ', HOSP_CHG_DESC) AS SOURCE_VALUE FROM PATBILL A`
		`JOIN CHGMSTR B ON A.STD_CHG_CODE=B.STD_CHG_CODE`
		`JOIN hospchg C ON A.hosp_chg_id=C.hosp_chg_id`
		`) A`
		`UNION`
		`(`
		`SELECT CPT_CODE AS SOURCE_VALUE FROM PATCPT`
		`)`
		For operation time records, NULL for now	

MEASUREMENT_SOURCE_CONCEPT_ID	-	QUERY: SOURCE TO SOURCE
		`SELECT SOURCE_CONCEPT_ID FROM CTE_VOCAB_MAP WHERE SOURCE_VOCABULARY_ID IN ('CPT4', 'HCPCS')`
		`AND TARGET_VOCABULARY_ID IN ('CPT4', 'HCPCS') AND DOMAIN_ID='Measurement'`
			Only populated for standard coding CPT4, and HCPCS codes

UNIT_SOURCE_VALUE	-	NULL	
VALUE_SOURCE_VALUE	-	NULL	

Example SQL (Redshift) for calculating surgical operating times:

```
WITH cte AS
(
  SELECT DISTINCT pp.pat_key,
         pp.icd_code,
         pp.proc_day,
         SUM(pb.std_qty *
         (CASE
            WHEN cm.std_chg_desc LIKE '%HR%' OR cm.std_chg_desc LIKE '%HOUR%'
            THEN 60*SPLIT_PART(TRIM(' ' FROM
REGEXP_REPLACE(cm.std_chg_desc,'[[:alpha:]]')),' ',1)
            ELSE 0
          END +
          CASE
            WHEN cm.std_chg_desc LIKE '%HR%MIN%'
            THEN 1 * SPLIT_PART(TRIM(' ' FROM
REGEXP_REPLACE(cm.std_chg_desc,'[[:alpha:]]')),' ',2)
            WHEN cm.std_chg_desc NOT LIKE '%HR%' AND cm.std_chg_desc LIKE
'%MIN%'
            THEN 1*RIGHT (TRIM(' ' FROM
REGEXP_REPLACE(cm.std_chg_desc,'[[:alpha:]]')),2)
            ELSE 0
          END)) AS total_mins
  FROM native.paticd_proc pp
  JOIN native.patbill pb
    ON pp.pat_key = pb.pat_key
    AND pp.proc_day = pb.serv_day
  JOIN native.chgmstr cm
    ON pb.std_chg_code = cm.std_chg_code
    AND cm.std_chg_code IN (SELECT std_chg_code
                            FROM native.chgmstr
                            WHERE clin_sum_desc = 'SURGERY TIME' AND
std_chg_code != 360360000530008) --hardcoding out std_chg_desc='OR MINOR FLAT
RATE', i.e. no associated time
  GROUP BY pp.pat_key,
           pp.icd_code,
           pp.proc_day
)
SELECT pat_key,
       proc_day,
       icd_code,
       SUM(total_mins)
FROM cte
GROUP BY pat_key,
         proc_day,
         icd_code
ORDER BY pat_key
```

3.1.10 TABLE NAME: DRUG_EXPOSURE

The DRUG_EXPOSURE table will house records from ***PATBILL*** and ***PATCPT*** that have been mapped to the drug or metadata domain.

Administrations of drugs are recorded in the ***PATBILL*** table as standard charges. Premier captures the day of administration in the ***SERV_DAY*** field. DRUG_EXPOSURE_START_DATE is determined by adding the number of service days to the visit start day using VISIT_OCCURRENCE .VISIT_START_DATE and ***PATBILL.SERV_DAY.*** If the start date is greater than the end of the month, then it's truncated to the end of month. Procedure drugs reside in the ***PATCPT*** table. DRUG_EXPOSURE_START_DATE for procedures is the last day of visit or VISIT_END_DATE, since dates for the administration of procedure drugs is not recorded, the assumption is made that the procedure occurred sometime before the end of the visit. DRUG_EXPOSURE_END_DATE cannot be determined because the patient is not followed each stay and days' supply information is not available.

Premier does not provide NDC codes for drugs that are administered during a visit. The PRESCRIBING_PROVIDER_ID is determined from the visit using the admitting provider id of the visit. In Premier, the admitting and attending providers are provided and due to the similarity of both the fields, the admitting provider id is used. The determination cannot be made if the admitting provider was the provider that prescribed the medication but that is the only information that is available. Drug type is considered inpatient administration for all drugs, except those drugs that are procedures and come from ***PATCPT***. Both HCPCS codes and CPT codes are available in ***PATCPT***. The quantity of drugs administered as captured from the ***QTY*** field in ***PATBILL***.

Each standard charge from PATBILL maps to the ***CHGMSTR*** table that houses additional information regarded the department, and descriptions about the item. To map each drug to an appropriate concept, USAGI was used to map ***STD_CHG_DESC*** to the value of an RxNorm concept. The ***CHGMSTR*** table is segmented by the department ***STD_DEPT_DESC***. The drug records represented in this table are captured from the Pharmacy department. Any mapping that cannot be correctly identified is mapped to a CONCEPT_ID of zero. All drugs will be mapped and included in this table even if they don't have a valid concept. All charges are loaded into the SOURCE_TO_CONCEPT_MAP table, and the table is attached in the appendix. The ***STD_CHG_CODE*** is mapped to a ***HOSP_CHG*** using the table ***HOSPCHG***, and each ***HOSP_CHG*** has a description that is displayed in the CDM as the source value.

Only records that fall within an OBSERVATION_PERIOD are available for each person. The VISIT_OCCURRENCE table must be created before the DRUG_EXPOSURE table is created.

The field mapping is performed as follows:

Destination Field	Source Field	Applied Rule	Comment
DRUG_EXPOSURE_ID	-	System generated	
PERSON_ID	***PAT.MEDREC_KEY***		
DRUG_CONCEPT_ID	***PATCPT.CPT_CODE*** ***PATBILL.STD_CHG_CODE***	QUERY: SOURCE TO STANDARD SELECT TARGET_CONCEPT_ID FROM CTE_VOCAB_MAP WHERE SOURCE_VOCABULARY_ID IN ('CPT4', 'HCPCS', 'JNJ_PMR_DRUG_CHRG_CD') AND TARGET_DOMAIN_ID = 'Drug'	Include all concepts that map to a concept id of zero.
DRUG_EXPOSURE_START_DATE	***PATBILL.SERV_DAY*** **VISIT_OCCURRENCE.VISIT_START_DATE** ***Or*** **VISIT_OCCURRENCE.VISIT_END_DATE**	If drug is from ***PATBILL*** use a combination of service day and visit start date unless the service day is greater than the end of the month If drug comes from ***PATCPT*** then use visit end date	
DRUG_EXPOSURE_START_DATETIME	-	NULL	
DRUG_EXPOSURE_END_DATE	**DRUG_EXPOSURE.DRUG_EXPOSURE_START_DATE**	**DRUG_EXPOSURE.DRUG_EXPOSURE_START_DATE**	Now a required field. No info on days supply, so set same date as drug_exposure_start_date
DRUG_EXPSURE_END_DATETIME	-	NULL	
VERBATIM_END_DATE	-	NULL	
DRUG_TYPE_CONCEPT_ID		38000180- Inpatient administration	
STOP_REASON	-	NULL	
REFILLS	-	NULL	
QUANTITY	***PATBILL.STD_QTY***		Value is applied only to records that come from PATBILL, else records from PATCPT or PATICD are NULL

Destination Field	Source Field	Applied Rule	Comment
DAYS_SUPPLY	-	NULL	
SIG	-	NULL	
ROUTE_CONCEPT_ID	-	NULL	
LOT_NUMBER	-	NULL	
PROVIDER_ID	***PAT.ADMPHY***	NULL	
VISIT_OCCURRENCE_ID	***PAT.PAT_KEY***		
DRUG_SOURCE_VALUE		SELECT SOURCE_VALUE FROM (SELECT CONCAT(STD_CHG_DESC, ' / ', HOSP_CHG_DESC) AS SOURCE_VALUE FROM PATBILL A JOIN CHGMSTR B ON A.STD_CHG_CODE=B.STD_CHG_CODE JOIN hospchg C ON A.hosp_chg_id=C.hosp_chg_id) A	
DRUG_SOURCE_CONCEPT_ID	-	NULL	
ROUTE_SOURCE_VALUE	-	NULL	
DOSE_UNIT_SOURCE_VALUE	-	NULL	

3.1.11 TABLE NAME: CONDITION_OCCURRENCE

The CONDITION_OCCURRENCE table will house records from ***PATBILL*** and ***PATICD_DIAG*** that have been mapped to the condition domain and SNOMED vocabulary.

Condition occurrences in Premier are stored in ***PATICD_DIAG*** as diagnosis codes. The table houses admitting, primary and secondary diagnosis by visit. The CDM transformation captures all 3 types of diagnoses. In many cases patients will have the same admitting and primary diagnosis. The field ***PATICD_DIAG.ICD_VERSION*** identifies the diagnosis code as either ICD-9 or ICD-10. The data contain ICD-9 codes for diagnoses prior to 2015/10/01 and ICD-10 codes for diagnoses on or after 2015/10/01. The condition start date is determined as the visit start date from the VISIT_OCCURRENCE table. The exact day of diagnosis is not recorded in Premier, thus the assumption is made that the diagnosis is made on the VISIT_START_DATE. The CONDITION_END_DATE is null because in Premier we are unaware of when the condition is no longer relevant to the patient. The ASSOCIATED_PROVIDER_ID that is provided is the randomly generated key provided by Premier for the provider that admitted the patient. There are two providers that exist in Premier, the admitting and attending. This ETL makes the decision to use admitting because it is unknown whether the admitting provider, attending provider or another person diagnosed the person.

The field mapping is performed as follows:

Destination Field	Source Field	Applied Rule	Comment
CONDITION_OCCURRENCE_ID	-	System-generated	
PERSON_ID	***PAT.MEDREC_KEY***		
CONDITION_CONCEPT_ID	***PATICD_DIAG.ICD_CODE*** ***PATBILL.STD_CHG_CODE***	**For records from PATBILL.STD_CHG_CODE:** **QUERY: SOURCE TO STANDARD** SELECT TARGET_VOCABULARY_ID FROM CTE_VOCAB_MAP WHERE SOURCE_VOCABULARY_ID IN ('JNJ_PMR_PROC_CHRG_CD') AND TARGET_DOMAIN_ID = 'Condition' **For records from PATICD_DIAG.ICD_CODE:** **where ICD_VERSION=9** QUERY: SOURCE TO STANDARD SELECT TARGET_CONCEPT_ID FROM CTE_VOCAB_MAP WHERE SOURCE_VOCABULARY_ID IN ('ICD9CM') AND TARGET_DOMAIN_ID = 'Condition' **For records from PATICD_DIAG.ICD_CODE:** **where ICD_VERSION=10** QUERY: SOURCE TO STANDARD SELECT TARGET_CONCEPT_ID FROM CTE_VOCAB_MAP WHERE SOURCE_VOCABULARY_ID IN ('ICD10CM') AND TARGET_DOMAIN_ID = 'Condition'	ICD9 diagnosis codes are mapped to SNOMED concepts
CONDITION_START_DATE	***PATBILL.SERV_DAY*** **VISIT_OCCURRENCE.VISIT_START_DATE** ***OR*** ***VISIT_OCCURRENCE.VISIT_START_DATE***	If condition is from ***PATBILL*** use a combination of service day and visit start date unless the service day is greater than the end of the month If observation comes from **PATICD_DIAG.ICD_CODE** then use visit start date	
CONDITION_START_DATETIME	-	NULL	
CONDITION_END_DATE	-	NULL	

Destination Field	Source Field	Applied Rule	Comment
CONDITION_END_DATETIME	-	NULL	
CONDITION_TYPE_CONCEPT_ID	-	**For records from PATICD_DIAG.ICD_CODE*:*** when ***PAT.I_O_IND*** ='I' and ***ICD.PRI_SEC*** ='P' then 38000183 when ***PAT.I_O_IND***='I' and ***PATICD_DIAG.ICD_PRI_SEC*** ='S' then 38000185 when ***PAT.I_O_IND***='O' and ***PATICD_DIAG.ICD_PRI_SEC*** ='P' then 38000215 when ***PAT.I_O_IND***='O' and ***PATICD_DIAG.ICD_PRI_SEC*** ='S' then 38000216 when ***PAT.I_O_IND*** in ('I' , "O") and ***PATICD_DIAG.ICD_PRI_SEC*** ='A' then 4203942 **For records from PATBILL.STD_CHG_CODE:** when ***PAT.I_O_IND*** ='I' then 38000186 when ***PAT.I_O_IND***='O' then 38000217	
STOP_REASON	-	NULL	
PROVIDER_ID	***PAT.ADMPHY***	NULL	
VISIT_OCCURRENCE_ID	***PAT.PAT_KEY***		
CONDITION_SOURCE_VALUE	***PATICD_DIAG.ICD_CODE***		

CONDITION_SOURCE_CONCEPT_ID		QUERY: SOURCE TO SOURCE
		SELECT SOURCE_CONCEPT_ID
		FROM CTE_VOCAB_MAP
		WHERE SOURCE_VOCABULARY_ID IN ('ICD9CM', 'ICD10', 'ICD10CM')
		AND TARGET_VOCABULARY_ID IN ('ICD9CM', 'ICD10', 'ICD10CM')
		AND DOMAIN_ID='CONDITION'	

CONDITION_STATUS_SOURCE_VALUE	***PATICD_DIAG.ICD_POA***		
CONDITION_STATUS_CONCEPT_ID	***PATICD_DIAG.ICD_POA***	**When PATICD_DIAG.ICD_POA in ('W', 'Y') then 46236988** **Else** **0**	

3.1.12 Table Name: PROCEDURE_OCCURRENCE

The PROCEDURE_OCCURRENCE table will house records from ***PATBILL***, ***PATCPT,*** and ***PATICD_PROC.*** Procedure records from ***PATBILL*** are mapped to the procedure domain; procedure records from ***PATCPT*** and procedure records from ***PATICD_PROC*** are mapped to the SNOMED vocabulary.

The ***PATBILL*** table holds all charges that were consumed within a visit. For our CDM, the drugs are separated and inserted into the DRUG_EXPOSURE table, and all other billing records are entered into the PROCEDURE_OCCURRENCE table. For records that are obtained through ***PATBILL***, the start date is determined from the service day in ***PATBILL*** and VISIT_START_DATE. If the combination of start date and service day records result in a date greater than the end of the month, the VISIT_END_DATE is assigned.

PATCPT houses the HCPCS and CPT codes by visit, and it is unknown when the procedure was performed. Procedure drugs are recorded as procedure drugs and move to the DRUG_EXPOSURE table. The procedure start date is identified as the VISIT_END_DATE from VISIT_OCCURRENCE. Procedure type is determined by the indicator of whether it was an inpatient stay or outpatient stay.

PATICD_PROC holds procedure codes that move to the PROCEDURE_OCCURRENCE table. The day the procedure was performed during the visit is recorded as ***PATICD_PROC.PROC_DAY.*** The PROCEDURE_DATE is determined by VISIT_START_DATE + PROC_DAY. If the combination of start date and procedure day records result in a date greater than the end of the month, the VISIT_END_DATE is assigned.

In order to map each drug to an appropriate concept, USAGI was used on the ***STD_CHG_DESC*** to map the value to a concept; all concepts that map into the procedure domain are included in this table. The ***STD_CHG_CODE*** is mapped to a ***HOSP_CHG*** using ***HOSPCHG***, and each ***HOSP_CHG*** has a description that is displayed in the CDM along with the standard change code descriptions. Billing records that do not map to a target concept are moved to PROCEDURE_OCCURRENCE with CONCEPT_ID=0.

Many CPT-4, CPT-4 Category III, and “C” HCPCS codes are embedded in Premier STD_CHG_CODES. Most CPT-4 codes do have a corresponding Premier standard charge item(s). If a CPT-4 code is embedded in a Premier standard charge item, then it will be in positions 7-11. Not every item on a hospital’s charge master, however, can be represented by a CPT-4 code. Examples would be items billed for pharmacy, room charges, central supplies, etc. Many “C” HCPCS codes (unique temporary pricing codes established by CMS for hospital outpatient department services and procedures) and CPT-4 Category III codes (temporary codes for emerging technology, services and procedures) are also embedded in the Premier standard charge code. For C codes, the C is dropped and replaced with a 0. For example, positions 7 – 11 of the standard charge code for embedded C code, C8921, is 08921. For temporary codes, the trailing T is dropped and the year it was created is tacked to the end. For example, for standard charge code 360360000192002, the CPT code is 0019T, and the year it was added is 2002. The CPT code, less the trailing T, is in positions 8 – 11, and the year is in positions 12 – 15 of the standard charge code. See the query below for extracting embedded codes from STD_CHG_CODE.

TODO: Procedure providers, ***PATICD_PROC.PROC_PHY***, are associated with procedure records from ***PATICD_PROC***. Procedure providers will be associated with PROCEDURE_OCCURRENCE records only. Procedure providers will also move to the PROVIDER table with an associated ***PROCPHY_SPEC***. Often, the procedure physician and admitting physician are the same person (ADM_PHY = PROC_PHY).

Records that have a valid OBSERVATION_PERIOD for each patient are included.

The field mapping is performed as follows:

Destination Field	Source Field	Applied Rule	Comment
PROCEDURE_OCCURRENCE_ID	-	System-generated	
PERSON_ID	***PAT.MEDREC_KEY***		

PROCEDURE_CONCEPT_ID	***PATBILL.STD_CHG_CODE***
	PATICD_PROC.ICD_CODE
	PATCPT.CPT_CODE
		QUERY SOURCE TO STANDARD:
		SELECT TARGET_CONCEPT_ID WHERE SOURCE_VOCABULARY_ID IN ('JNJ_PMR_PROC_CHRG_CD', 'CPT4', 'HCPCS', 'ICD10CM', 'ICD10PCS', 'ICD9CM', 'ICD9Proc') AND TARGET_DOMAIN_ID ='Procedure' AND SOURCE_CONCEPT_CLASS_ID NOT IN ('CPT4 Modifier', 'ICD10PCS Hierarchy')
		SELECT TARGET_CONCEPT_ID FROM CTE_VOCAB_MAP WHERE SOURCE_VOCABULARY_ID IN ('JNJ_PMR_PROC_CHRG_CD' AND TARGET_CONCEPT_ID=0	

PROCEDURE_DATE	***VISIT_OCCURRENCE.VISIT_END_DATE*** or ***VISIT_OCCURRENCE.VISIT_START_DATE*** ***PATBILL.SERV_DAY*** or ***VISIT_OCCURRENCE.VISIT_START_DATE*** ***PATICD_PROC.PROC_DAY***		If the procedure is a CPT code then discharge date is used as procedure date because the exact date is unknown. If the row is coming from PATBILL then a combination or admit date and service date is used. If the record comes from PATICD_PROC then a combination of admit date and service date is used.

PROCEDURE_DATETIME	-	NULL	
PROCEDURE_TYPE_CONCEPT_ID	***PATICD_PROC.ICD_PRI_SEC***	When ***PATICD_PROC.ICD_PRI_SEC*** ='P' then 44786630 When ***PATICD_PROC.ICD_PRI_SEC*** ='S' then 44786631	All CHGMSTR procedures will be assigned a PROCEDURE_TYPE_CONCEPT_ID indicating 1st position.
MODIFIER_CONCEPT_ID	-	NULL	
QUANTITY	PATBILL.STD_QTY	Quantities are populated for all records obtained from the billing record.	
PROVIDER_ID	***PATICD_PROC.PROC_PHY***		
VISIT_OCCURRENCE_ID	***PAT.PAT_KEY***		

PROCEDURE_SOURCE_VALUE	***PATICD_PROC.ICD CODE***
	Or
	PATCPT.CPT_CODE
	For all other procedures:
	CHGMSTR.STD_CHG_CODE_DESC/ HOSP_CHG.HOSP_CHG_DESC
		`SELECT SOURCE_VALUE FROM`
		`(`
		`SELECT CONCAT(STD_CHG_DESC, ' / ', HOSP_CHG_DESC) AS SOURCE_VALUE FROM PATBILL A`
		`JOIN CHGMSTR B ON A.STD_CHG_CODE=B.STD_CHG_CODE`
		`JOIN hospchg C ON A.hosp_chg_id=C.hosp_chg_id`
		`) A`
		
		`SELECT SOURCE_VALUE FROM`
		`(`
		`SELECT ICD_CODE FROM PATICD_PROC A`
		`JOIN CONCEPT C ON C.CONCEPT_CODE=A.ICD_CODE`
		`WHERE VOCABULARY_ID='ICDProc'`
		`) A`
		
		`UNION`
		
		`(`
		`SELECT CPT_CODE AS SOURCE_VALUE FROM PATCPT`
		`)`
			To preserve the most detailed description of procedures, if hospital charge descriptions are available, they are to be used, otherwise standard charge code description is displayed

PROCEDURE_SOURCE_CONCEPT_ID	-	SELECT SOURCE_CONCEPT_ID FROM CTE_VOCAB_MAP WHERE SOURCE_VOCABULARY_ID IN ('ICD9Proc', 'CPT4', 'HCPCS') AND TARGET_VOCABULARY_ID IN ('ICD9Proc', 'CPT4', 'HCPCS') AND DOMAIN_ID='Procedure'
		SELECT SOURCE_VALUE FROM (SELECT CONCAT(STD_CHG_DESC, ' / ', HOSP_CHG_DESC) AS SOURCE_VALUE FROM PATBILL A JOIN CHGMSTR B ON A.STD_CHG_CODE=B.STD_CHG_CODE JOIN hospchg C ON A.hosp_chg_id=C.hosp_chg_id) A	

QUALIFER_SOURCE_VALUE	-	NULL	

Example SQL (Redshift) for extracting CPT4, CPT4-3, and HCPCS codes from STD_CHG_CODE for vocabulary mapping:

```
WITH CTE_CPT4 AS (
  SELECT CONCEPT_CODE AS FIXED_CONCEPT_CODE, CONCEPT_NAME, CONCEPT_ID,
DOMAIN_ID, CONCEPT_CODE, VOCABULARY_ID
      FROM CDM.CONCEPT
      WHERE VOCABULARY_ID = 'CPT4'
      AND CONCEPT_CLASS_ID = 'CPT4'
      AND STANDARD_CONCEPT = 'S'
),
CTE_HCPCS AS (
  SELECT CONCAT('0',SUBSTRING(CONCEPT_CODE,2,4)) AS FIXED_CONCEPT_CODE,
CONCEPT_ID, CONCEPT_NAME, DOMAIN_ID, CONCEPT_CODE, VOCABULARY_ID
      FROM CDM.CONCEPT
      WHERE VOCABULARY_ID = 'HCPCS'
      AND SUBSTRING(CONCEPT_CODE,1,1) = 'C'
      AND STANDARD_CONCEPT = 'S'
),
CTE_CPT4_3 AS (
  SELECT SUBSTRING(CONCEPT_CODE,1,4) AS FIXED_CONCEPT_CODE, CONCEPT_NAME,
CONCEPT_ID, DOMAIN_ID, CONCEPT_CODE, VOCABULARY_ID
      FROM CDM.CONCEPT
      WHERE VOCABULARY_ID = 'CPT4'
      AND CONCEPT_CLASS_ID = 'CPT4'
      AND STANDARD_CONCEPT = 'S'
      AND SUBSTRING(CONCEPT_CODE,5,1) = 'T'
),
CTE_CODE_PULL AS (
  SELECT
    CASE
      WHEN c1.CONCEPT_ID IS NOT NULL THEN c1.CONCEPT_ID
      WHEN c2.CONCEPT_ID IS NOT NULL AND SUM_DEPT_DESC NOT IN ('SUPPLY',
'PHARMACY') THEN c2.CONCEPT_ID
      WHEN c3.CONCEPT_ID IS NOT NULL THEN c3.CONCEPT_ID
      WHEN c4.CONCEPT_ID IS NOT NULL THEN c4.CONCEPT_ID
      ELSE NULL
    END TARGET_CONCEPT_ID,
    CASE
      WHEN c1.CONCEPT_NAME IS NOT NULL THEN c1.CONCEPT_NAME
      WHEN c2.CONCEPT_NAME IS NOT NULL AND SUM_DEPT_DESC NOT IN ('SUPPLY',
'PHARMACY') THEN c2.CONCEPT_NAME
      WHEN c3.CONCEPT_NAME IS NOT NULL THEN c3.CONCEPT_NAME
      WHEN c4.CONCEPT_NAME IS NOT NULL THEN c4.CONCEPT_NAME
      ELSE NULL
    END TARGET_CONCEPT_NAME,
    CASE
      WHEN c1.CONCEPT_CODE IS NOT NULL THEN c1.CONCEPT_CODE
      WHEN c2.CONCEPT_CODE IS NOT NULL AND SUM_DEPT_DESC NOT IN ('SUPPLY',
'PHARMACY') THEN c2.CONCEPT_CODE
      WHEN c3.CONCEPT_CODE IS NOT NULL THEN c3.CONCEPT_CODE
      WHEN c4.CONCEPT_CODE IS NOT NULL THEN c4.CONCEPT_CODE
      ELSE NULL
    END TARGET_CONCEPT_CODE,
    CASE
      WHEN c1.VOCABULARY_ID IS NOT NULL THEN c1.VOCABULARY_ID
      WHEN c2.VOCABULARY_ID IS NOT NULL AND SUM_DEPT_DESC NOT IN ('SUPPLY',
'PHARMACY') THEN c2.VOCABULARY_ID
      WHEN c3.VOCABULARY_ID IS NOT NULL THEN c3.VOCABULARY_ID
      WHEN c4.VOCABULARY_ID IS NOT NULL THEN c4.VOCABULARY_ID
      ELSE NULL
    END TARGET_VOCABULARY_ID,
    CASE
      WHEN c1.DOMAIN_ID IS NOT NULL THEN c1.DOMAIN_ID
      WHEN c2.DOMAIN_ID IS NOT NULL AND SUM_DEPT_DESC NOT IN ('SUPPLY',
'PHARMACY') THEN c2.DOMAIN_ID
      WHEN c3.DOMAIN_ID IS NOT NULL THEN c3.DOMAIN_ID
      WHEN c4.DOMAIN_ID IS NOT NULL THEN c4.DOMAIN_ID
      ELSE NULL
    END TARGET_DOMAIN_ID,
    CASE
      WHEN c1.CONCEPT_ID IS NOT NULL THEN '1-CPT4'
      WHEN c2.CONCEPT_ID IS NOT NULL AND SUM_DEPT_DESC NOT IN ('SUPPLY',
'PHARMACY') THEN '2-HCPCs'
      WHEN c3.CONCEPT_ID IS NOT NULL THEN '3-CPT4 III'
      WHEN c4.CONCEPT_ID IS NOT NULL THEN '4-USAGI Mapping'
      ELSE '5-UNMAPPED'
    END TARGET_FLAG,
    cm.*
  FROM CHGMSTR cm
    LEFT OUTER JOIN CTE_CPT4 c1
```

```
      ON c1.FIXED_CONCEPT_CODE = SUBSTRING(cm.STD_CHG_CODE,7,5)
    LEFT OUTER JOIN CTE_HCPCS c2
      ON c2.FIXED_CONCEPT_CODE = SUBSTRING(cm.STD_CHG_CODE,7,5)
    LEFT OUTER JOIN CTE_CPT4_3 c3
      ON c3.FIXED_CONCEPT_CODE = substring(std_chg_code, 8, 4)
      AND  substring(std_chg_code, 12, 4) in
      ('2000', '2001', '2002', '2003', '2004', '2005', '2006', '2007',
'2008',
       '2009', '2010', '2011', '2012', '2013', '2014', '2015', '2016',
'2017')
    LEFT OUTER JOIN cdm.SOURCE_TO_CONCEPT_MAP stcm
      ON stcm.SOURCE_CODE = cm.STD_CHG_CODE
      AND SOURCE_VOCABULARY_ID IN (
      'JNJ_PMR_DRUG_CHRG_CD','JNJ_PMR_PROC_CHRG_CD'
      )
      AND stcm.TARGET_CONCEPT_ID != 0
    LEFT OUTER JOIN cdm.CONCEPT c4
      ON stcm.TARGET_CONCEPT_ID = c4.CONCEPT_ID
)
SELECT DISTINCT cp.*,
  CASE WHEN z.CODE_COUNT IS NULL THEN 0 ELSE z.CODE_COUNT END AS CODE_COUNT
FROM CTE_CODE_PULL cp
  LEFT OUTER JOIN (
    SELECT STD_CHG_CODE, COUNT(*) AS CODE_COUNT
    FROM PATBILL
    GROUP BY STD_CHG_CODE
   ) z
    ON z.STD_CHG_CODE = cp.STD_CHG_CODE
```

3.1.13 TABLE NAME: DEVICE_EXPOSURE

The DEVICE_EXPOSURE table will house records from ***PATBILL, PATCPT, PATICD_DIAG, and PATCID_PROC*** that have been mapped to the device domain.

The Premier database has information captured about the various devices in the billing table PATBILL and PATCPT includes codes that are mapped into the device domain. USAGI is used to make PATBILL records to standard concepts, and concepts that are in the Device domain are included. Similarly, HCPCS codes extracted from STD_CHG_CODE records that map to the Device domain also move to DEVICE_EXPOSURE. The ASSOCIATED_PROVIDER_ID that is provided is the randomly generated key provided by Premier for the provider that admitted the patient. There are two providers that exist in Premier, the admitting and attending. This ETL makes the decision to use admitting because it is unknown whether the admitting provider, attending provider or another person diagnosed the person. The ***STD_CHG_CODE*** is mapped to a ***HOSP_CHG*** using ***HOSPCHG***, and each ***HOSP_CHG*** has a description that is displayed in the CDM along with the standard change code descriptions.

Many CPT-4, CPT-4 Category III, and “C” HCPCS codes are embedded in Premier STD_CHG_CODES. Most CPT-4 codes do have a corresponding Premier standard charge item(s). If a CPT-4 code is embedded in a Premier standard charge item, then it will be in positions 7-11. Not every item on a hospital’s charge master, however, can be represented by a CPT-4 code. Examples would be items billed for pharmacy, room charges, central supplies, etc. Many “C” HCPCS codes (unique temporary pricing codes established by CMS for hospital outpatient department services and procedures) and CPT-4 Category III codes (temporary codes for emerging technology, services and procedures) are also embedded in the Premier standard charge code. For C codes, the C is dropped and replaced with a 0. For example, positions 7 – 11 of the standard charge code for embedded C code, C8921, is 08921. For temporary codes, the trailing T is dropped and the year it was created is tacked to the end. For example, for standard charge code 360360000192002, the CPT code is 0019T, and the year it was added is 2002. The CPT code, less the trailing T, is in positions 8 – 11, and the year is in positions 12 – 15 of the standard charge code. See the query in PROCEDURE_OCCURRENCE section for extracting embedded codes from STD_CHG_CODE.

Records that have a valid OBSERVATION_PERIOD for each patient are included.

The field mapping is as follows:

Destination Field	Source Field	Applied Rule	Comment
DEVICE_EXPOSURE_ID	-	System-generated	
PERSON_ID	***PAT.MEDREC_KEY***		

DEVICE_CONCEPT_ID	***PATBILL.STD_CHG_CODE***
	PATICD_PROC.ICD_CODE
	PATICD_DIAG.ICD_CODE
	PATCPT.CPT_CODE
		`QUERY:SOURCE To STANDARD:`
		`SELECT TARGET_CONCEPT_ID FROM CTE_VOCAB_MAP WHERE SOURCE_VOCABULARY_ID IN ('HCPCS', 'ICD10CM', 'JNJ_PMR_PROC_CHRG_CD') AND TARGET_DOMAIN_ID IN ('Device')`	

Destination Field	Source Field	Applied Rule	Comment
DEVICE_EXPOSURE_START_DATE	***VISIT_OCCURRENCE.VISIT_END_DATE*** or ***VISIT_OCCURRENCE.VISIT_START_DATE*** ***PATBILL.SERV_DAY***		If the device is a CPT code or HCPCS code then discharge date is used as device date because the exact date is unknown. If the row is coming from PATBILL then a combination or admit date and service date is used.
DEVICE_EXPOSURE_START_DATETIME	-	NULL	
DEVICE_EXPOSURE_END_DATE			
DEVICE_EXPOSURE_END_DATETIME	-	NULL	
DEVICE_TYPE_CONCEPT_ID	-	44818705 Inferred from procedure claim	
UNIQUE_DEVICE_ID	**-**	NULL	
PROVIDER_ID	***PAT.ADMPHY***		
VISIT_OCCURRENCE_ID	PAT.PAT_KEY		
DEVICE_SOURCE_VALUE	***PATCPT.CPT_CODE*** ***For all other procedures:*** ***CHGMSTR.STD_CHG_CODE_DESC/ HOSP_CHG.HOSP_CHG_DESC***	SELECT SOURCE_VALUE FROM (SELECT CONCAT(STD_CHG_DESC, ' / ', HOSP_CHG_DESC) AS SOURCE_VALUE FROM PATBILL A JOIN CHGMSTR B ON A.STD_CHG_CODE=B.STD_CHG_CODE JOIN hospchg C ON A.hosp_chg_id=C.hosp_chg_id) A UNION (SELECT CPT_CODE AS SOURCE_VALUE FROM PATCPT)	To preserve the most detailed description of procedures, if hospital charge descriptions are available, they are to be used, otherwise standard charge code description is displayed
DEVICE_SOURCE_CONCEPT_ID	-	NULL	

3.1.14 TABLE NAME: COST

Costs and charges are captured in Premier and are housed in the ***PATBILL*** table. Each record includes a billing item identified by ***STD_CHG_CODE*** and its associated charge to the patient (***BILL_CHARGES***) and cost to the provider (***BILL_COST***). Amounts paid (by patient, payer, copay, etc.) are not captured in Premier. Since each ***PATBILL*** record includes a cost and charge component, these records will be referred to as billing records henceforth. Each billing record is categorized by department using the REVENUE_CONCEPT_CODE_ID field.

Drug Costs: Drug exposure records that come from the ***PATBILL*** table will have costs associated with them, but drug exposure records that come from ***PATICD*** or ***PATCPT*** will not have an associated cost. BILL_COST and BILL_CHARGES for DRUG_EXPOSURE records that originate from ***PATBILL*** will be stored in the COST table. Use DRUG_EXPOSURE logic to determine which ***PATBILL*** records are drug exposures.

Procedure Costs: Procedure occurrence records that come from the ***PATBILL*** table will have costs associated with them, but procedure occurrence records that come from ***PATICD_PROC*** or ***PATCPT*** will not have an associated cost. BILL_COST and BILL_CHARGES for PROCEDURE_OCCURRENCE records that originate from ***PATBILL*** will be stored in the COST table. Use PROCEDURE_OCCURRENCE logic to determine which ***PATBILL*** records are procedure occurrences.

Device Costs: Device exposure records that come from the ***PATBILL*** table will have costs associated with them, but device exposure records that come from ***PATICD_DIAG, PATICD_PROC,*** or ***PATCPT*** will not have an associated cost. BILL_COST and BILL_CHARGES for DEVICE_EXPOSURE records that originate from ***PATBILL*** will be stored in the COST table. Use DEVICE_EXPOSURE logic to determine which ***PATBILL*** records are device exposures.

Observation Costs: Observation records that come from the ***PATBILL*** table will have costs associated with them, but observation records that come from ***PAT.MS_DRG, PATICD_DIAG, PATICD_PROC,*** and ***PATCPT*** will not have an associated cost. BILL_COST and BILL_CHARGES for OBSERVATION records that originate from ***PATBILL*** will be stored in the COST table. Use OBSERVATION logic to determine which ***PATBILL*** records are observation records.

Measurement Costs: Measurement records that come from the ***PATBILL*** table will have costs associated with them, but measurement records that come from ***PATCPT*** will not have an associated cost. BILL_COST and BILL_CHARGES for MEASUREMENT records that originate from ***PATBILL*** will be stored in the COST table. Use MEASUREMENT logic to determine which ***PATBILL*** records are observation records.

DRG codes: DRG information is captured in ***PAT.MS_DRG***, where one code is associated with one visit. Although a single visit will have multiple associated cost records, the same DRG code will be associated with each cost record for the visit.

Note: MS_DRG codes contain 3 digits, including those codes with leading zeros (i.e. 00# and 0##). These leading zeros are missing in Premier data and must be added in the ETL process for accurate mapping using the source to standard cte_vocab_map.

The field mapping is as follows:

Destination Field	Source Field	Applied Rule	Comment
COST_ID		A unique identifier for each COST record	
COST_EVENT_ID	DRUG_EXPOSURE.DRUG_EXPOSURE_ID OR PROCEDURE_OCCURRENCE.PROCEDURE_OCCURRENCE_ID OR DEVICE_EXPOSURE.DEVICE_EXPOSURE_ID OR OBSERVATION.OBSERVATION_ID OR MEASUREMENT.MEASUREMENT_ID		
COST_DOMAIN_ID		WHEN cost_event_id = drug_exposure.drug_exposure_id THEN cost_domain_id = 'Drug' OR WHEN cost_event_id = procedure_occurrence.procedure_occurrence_id THEN cost_domain_id = 'Procedure' OR WHEN cost_event_id = device_exposure.device_exposure_id THEN cost_domain_id = 'Device' OR WHEN cost_event_id = observation.observation_id THEN cost_domain_id = 'Observation' OR WHEN cost_event_id = measurement.measurement_id THEN cost_domain_id = 'Measurement'	

Destination Field	Source Field	Applied Rule	Comment
COST_TYPE_CONCEPT_ID	NULL	Currently NULL but to adhere to standard decided upon here: http://forums.ohdsi.org/t/discrepancy-in-understanding-the-cost-type-concept-id/1805	
CURRENCY_CONCEPT_ID		44818668- American dollar	

TOTAL_CHARGE	PATBILL.BILL_CHARGES	SELECT bill_charges
		FROM patbill
			Note we are not moving total visit costs or charges from PAT.PAT_COST and PAT.PAT_CHARGES

TOTAL_COST	PATBILL.BILL_COST	SELECT bill_cost
		FROM patbill	

TOTAL_PAID		Null	
PAID_BY_PAYER		Null	
PAID_BY_PATIENT		Null	
PAID_PATIENT_COPAY		Null	
PAID_PATIENT_COINSURANCE		Null	
PAID_PATIENT_DEDUCTIBLE		Null	
PAID_BY_PRIMARY		Null	
PAID_INGREDIENT_COST		Null	
PAID_DISPENSING_FEE		Null	
PAYER_PLAN_PERIOD_ID		Null	
AMOUNT_ALLOWED		Null	

REVENUE_CODE_CONCEPT_ID		QUERY:SOURCE To STANDARD:
		
		SELECT TARGET_CONCEPT_ID
		FROM CTE_VOCAB_MAP
		WHERE SOURCE_VOCABULARY_ID IN (JNJ_PMR_COST_CHRG_CD')
		AND TARGET_DOMAIN_ID IN ('Revenue Code')	

REVENUE_CODE_SOURCE_VALUE	CHGMSTR.SUM_DEPT_DESC	SELECT sum_dept_desc || '/' || std_dept_desc AS revenue_code_source_value
		FROM chgmstr	

DRG_CONCEPT_ID	***PAT.MS_DRG***	QUERY: SOURCE TO STANDARD
		SELECT TARGET_CONCEPT_ID
		FROM CTE_VOCAB_MAP
		WHERE SOURCE_VOCABULARY_ID IN ('DRG')	

DRG_SOURCE_VALUE	***PAT.MS_DRG***		

3.1.15 TABLE NAME: NOTE

Describe how the NOTE mapping and transformation are designed.

Destination Field	Source Field	Applied Rule	Comment
NOTE_ID			
PERSON_ID			
NOTE_DATE			
NOTE_TIME			
NOTE_TYPE_CONCEPT_ID			
NOTE_CLASS_CONCEPT_ID			
NOTE_TITLE			
NOTE_TEXT			
ENCODING_CONCEPT_ID			
LANGUAGE_CONCEPT_ID			
PROVIDER_ID			
NOTE_SOURCE_VALUE			
VISIT_OCCURRENCE_ID			

3.1.16 TABLE NAME: PROVIDER

Premier does provide individual providers for each visit, and it houses the admitting and attending provider specialty for each visit. For Premier, each hospital will have a set of providers which will be identified with a unique key generated by Premier. Two providers will be indicated for each visit, an admitting provider, and an attending provider. In most cases these fields are the same, thus the assumption is made to use admitting provider for each stay. Each provider is linked to a specialty. In addition, each provider will be linked to a Premier hospital through the ***PROV_ID***. The mapping for the specialties from the ***PHYSPEC*** table to valid concepts is in the mapping table in the appendix. Any providers that are not listed in the look up tables are added manually and associated to an unknown provider. The mapping table uses the most closely matching concept value.

Procedure providers, ***PATICD_PROC.PROC_PHY***, are associated with procedure records from ***PATICD_PROC***. Procedure providers will be associated with PROCEDURE_OCCURRENCE records only. Procedure providers will also move to the PROVIDER table with an associated ***PROCPHY_SPEC***. Often, the procedure physician and admitting physician are the same person (ADM_PHY = PROC_PHY).

Admitting and attending physician keys are defaulted to 99999999 and physician specialty is set to 900 when these data are not provided by the hospital. Move PAT.ADM_PHY=99999999 to PROVIDER.PROVIDER_ID=99999999. Similarly, some providers have unknown specialty, admphy_spec=900, set these to PROVIDER.PROVIDER_ID=38004514.

The field mapping is as follows:

Destination Field	Source Field	Applied Rule	Comment
PROVIDER_ID	***PAT.ADM_PHY***	System generated unique code	
PROVIDER_NAME	**-**	NULL	
NPI	-	NULL	
DEA	-	NULL	
SPECIALTY_CONCEPT_ID		QUERY: SOURCE TO STANDARD `SELECT TARGET_CONCEPT_ID FROM CTE_VOCAB_MAP WHERE SOURCE_VOCABULARY_ID = 'JNJ_PMR_P_SPCLTY'`	
CARE_SITE_ID	***PAT.PROV_ID***		
YEAR_OF_BIRTH	**-**	NULL	
GENDER_CONCEPT_ID	**-**	NULL	
PROVIDER_SOURCE_VALUE	***PAT.ADMPHY_SPEC***		
SPECIALTY_SOURCE_VALUE	***PHYSPEC.PHY_SPEC_DESC***		
SPECIALITY_SOURCE_CONCEPT	**-**	NULL	

Destination Field	Source Field	Applied Rule	Comment
_ID			
GENDER_SOURCE_VALUE	-	NULL	
GENDER_SOURCE_CONCEPT_ID	-	NULL	

3.2 Source Independent Data Mapping

3.2.1 Table Name: DRUG_ERA

A Drug Era is defined as a span of time when the Person is assumed to be exposed to a 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. There will only be one type of persistence window (duration that can elapse between drug exposures) applied to this CDM, which is 30 days.

Drugs that are mapped to a DRUG_CONCEPT_ID=0 should not be mapped. The logic below is used to map DRUG_CONCEPT_ID's to ingredients.

```
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
            AND c2.CONCEPT_VOCABULARY_ID = 8
            AND c2.CONCEPT_LEVEL = 2
WHERE c1.CONCEPT_VOCABULARY_ID = 8
```

Do not include records that cannot be mapped to the ingredient level. The DRUG_EXPOSURE_END_DATE is the DRUG_EXPOSURE_START_DATE.

The field mapping is as follows:

Destination Field	Source Field	Applied Rule	Comment
DRUG_ERA_ID		System generated	
PERSON_ID	PERSON_ID		
DRUG_CONCEPT_ID	DRUG_CONCEPT_ID	Do no create DRUG_ERAs where the DRUG_EXPOSURE.DRUG_CONCEPT_ID is 0. Use the map above to map DRUG_EXPOSURE.DRUG_CONCEPT_ID to the ingredient level DRUG_CONCEPT_ID used in the DRUG_ERA.	
DRUG_ERA_START_DATE	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.START_DATE		
DRUG_TYPE_CONCEPT_ID	-	Apply a 30-day persistence window and label as CONCEPT_ID 38000182 (Drug era - 30 days persistence window).	Falls under CONCEPT_VOCABULARY_ID = 36 as a Drug Exposure Type.
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: CONDITION_ERA

CONDITION_ERAs are chronological periods of condition occurrence. There will only be one type of persistence window (duration that can elapse between condition occurrences) applied to this CDM, which is 30 days. CONDITION_END_DATE will be the CONDITION_START_DATE.

Exclude records with a CONDITION_CONCEPT_ID=0.

All Condition Eras are recorded in the CONDITION_ERA table based on the following field mapping:

Destination Field	Source Field	Applied Rule	Comment
CONDITION_ERA_ID		System-generated	
PERSON_ID	PERSON_ID		
CONDITION_CONCEPT_ID	CONDITION_CONCEPT_ID	Do not build condition_era where the condition_occurance_condition_concept_id=0	
CONDITION_ERA_START_DATE	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	CONDITION_START_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 (Condition era - 30 days persistence window).	Falls under CONCEPT_VOCABULARY_ID = 37 - OMOP Condition Occurrence Type.
CONDITION_OCCURRENCE_COUNT		Sum up the number of CONDITION_OCCURRENCEs for this PERSON_ID and this CONCEPT_ID during the exposure window being built.	

3.2.2 TABLE NAME: DOSE_ERA

A Dose Era is defined as a span of time when the Person is assumed to be exposed to a constant dose of a specific active ingredient.

Destination Field	Source Field	Applied Rule	Comment
DOSE_ERA_ID		System-generated	
PERSON_ID	PERSON_ID		
DRUG_CONCEPT_ID	DRUG_CONCEPT_ID	Do not build dose_era where the drug_concept_id=0	
UNIT_CONCEPT_ID			
DOSE_VALUE			Numeric value of dose
DOSE_ERA_START_DATE	DRUG_EXPOSURE_START_DATE		The start date for the dose era constructed from the individual instances of drug exposures. It is the

Destination Field	Source Field	Applied Rule	Comment
			start date of the very first chronologically recorded instance of utilization of a drug.
DOSE_ERA_END_DATE			

4.0 Appendix

Queries for VOCABULARY:

```
--SOURCE TO SOURCE
WITH CTE_VOCAB_MAP AS (
       SELECT c.concept_code AS SOURCE_CODE, c.concept_id AS
SOURCE_CONCEPT_ID, c.vocabulary_id AS SOURCE_VOCABULARY_ID, c.domain_id AS
SOURCE_DOMAIN_ID, c.concept_class_id AS SOURCE_CONCEPT_CLASS_ID,
                     c.invalid_reason AS SOURCE_INVALID_REASON,
                     c.concept_ID as TARGET_CONCEPT_ID, 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_vocabulary_id,
c1.domain_id AS SOURCE_DOMAIN_ID, c2.CONCEPT_CLASS_ID AS
SOURCE_CONCEPT_CLASS_ID,
                     stcm.INVALID_REASON AS
SOURCE_INVALID_REASON,target_concept_id, 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
)

--SOURCE TO STANDARD
WITH CTE_VOCAB_MAP AS (
       SELECT c.concept_code AS SOURCE_CODE, c.concept_id AS
SOURCE_CONCEPT_ID, c.vocabulary_id AS SOURCE_VOCABULARY_ID,
                           c.domain_id AS SOURCE_DOMAIN_ID,
c.CONCEPT_CLASS_ID AS SOURCE_CONCEPT_CLASS_ID, c.INVALID_REASON AS
SOURCE_INVALID_REASON,
                           c1.concept_id AS TARGET_CONCEPT_ID,
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_vocabulary_id,
c1.domain_id AS SOURCE_DOMAIN_ID, c2.CONCEPT_CLASS_ID AS
SOURCE_CONCEPT_CLASS_ID,
                           stcm.INVALID_REASON AS SOURCE_INVALID_REASON,
        target_concept_id, 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
)
```