A PUBLIC PRIVATE RESEARCH PARTNERSHIP OF THE FNIH

OBSERVATIONAL
MEDICAL
OUTCOMES
PARTNERSHIP

OMOP Common Data Model (CDM V4.0)
Health Cost and Utilization Project (HCUP) – Nationwide Inpatient Sample (NIS)
Mapping Specification

Version 1.1

8-January-2015

Martijn Schuemie

Erica Voss

Patrick Ryan

Chris Knoll

Source Data Mapping Approach

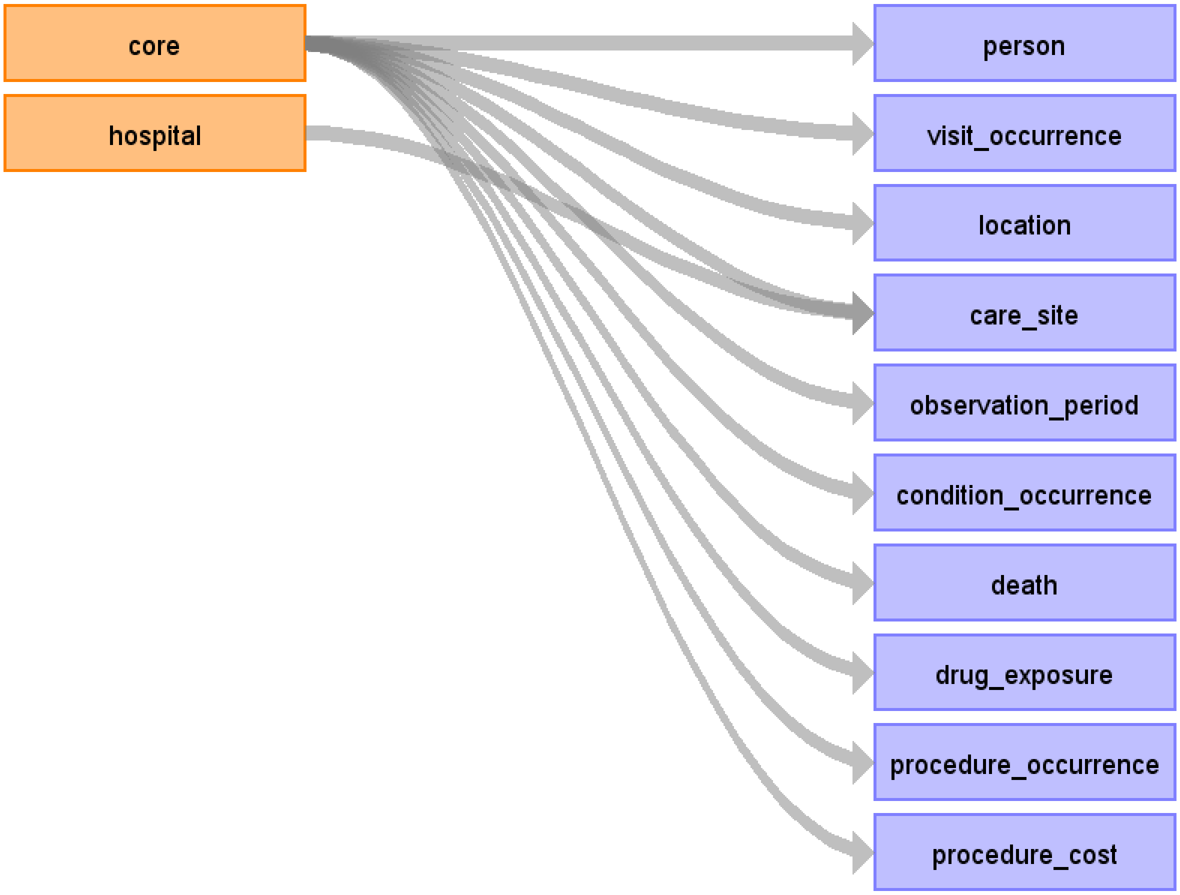
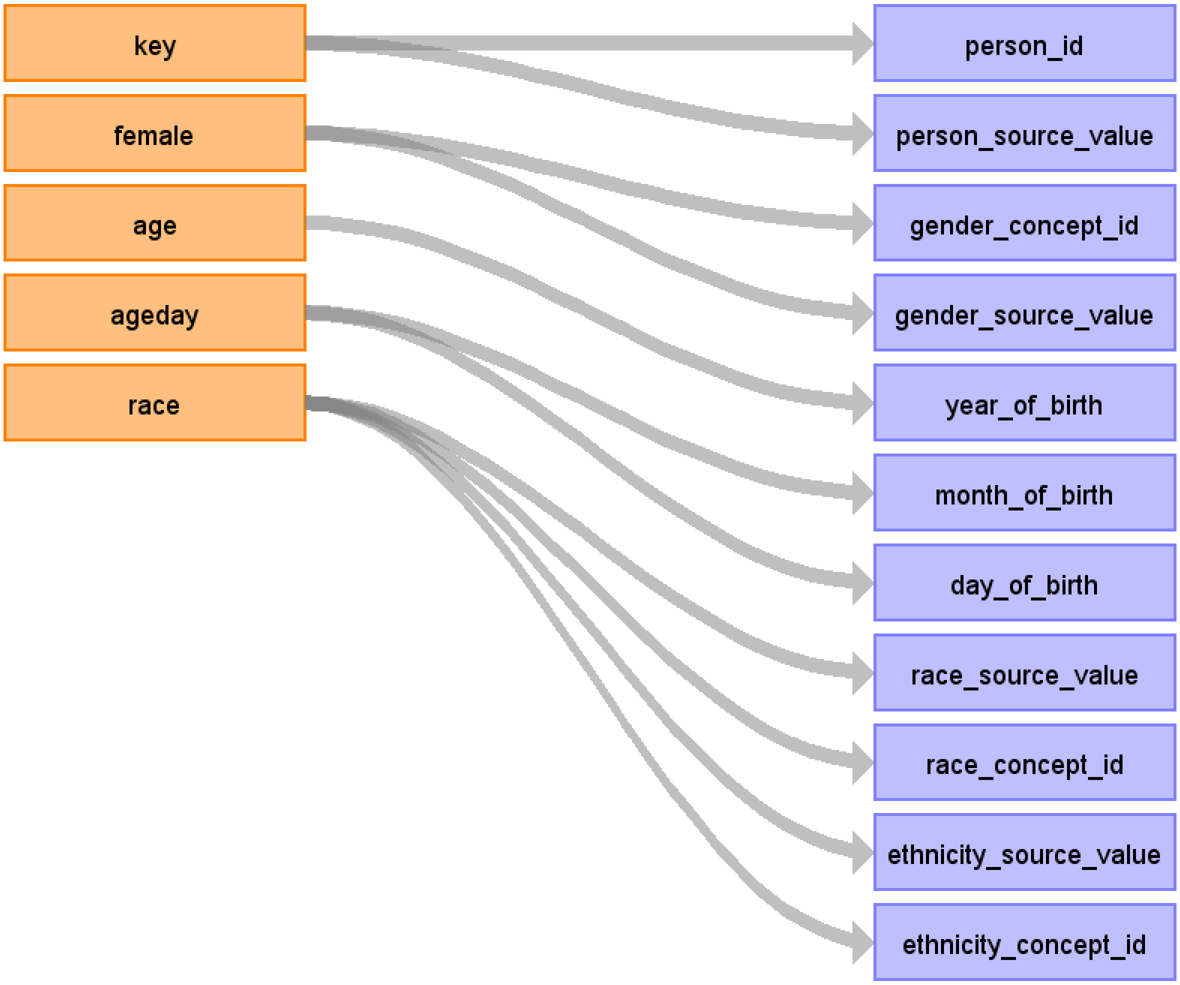


Table name: person

Reading from core

Contains hospital discharges, not persons. But every row can be considered one person



Destination Field	Source Field	Logic	Comment
person_id	key		One-on-one copy
person_source_value	key		
gender_concept_id	female	1 = female (8532) 0 = male (8507) else = unknown (8551)	
gender_source_value	female		

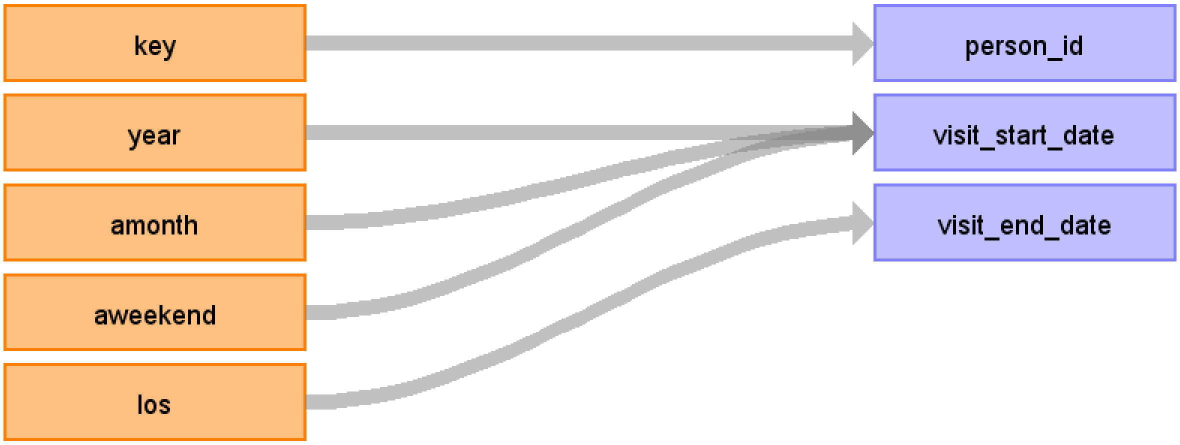
year_of_birth	age	if age > 0 year_of_birth = visit_start_date year - age else year_of_birth + month_of_birth + day_of_birth = visit_start_date - ageday	
month_of_birth	ageday	see year_of_birth logic. Only filled if age = 0, else NULL	
day_of_birth	ageday	see year_of_birth logic. Only filled if age = 0, else NULL	
race_source_value	race		
race_concept_id	race	See mapping table: source_value label concept_id 1 White 8527 2 Black 8516 3 Other 8522 4 Pacific Islander 8557 5 Native American 8657 6 Other 8522	
ethnicity_source_value	race	if race = 3, put that value here, else 0 (note: will be stored in race_source_value)	
ethnicity_concept_id	race	if race = 3, set to 38003563 (Hispanic), else 0	
location_id		not used	
provider_id		not used	
care_site_id		not used	

Sample code for transformation:

```
Row person = new Row();
person.add("person_id", row.get("KEY"));
person.add("person_source_value", row.get("KEY"));
person.add("gender_source_value", row.get("FEMALE"));
person.add("gender_concept_id", row.get("FEMALE").equals("1") ? "8532" :
row.get("FEMALE").equals("0") ? "8507" : "8551");

if (row.getInt("AGE") > 0) {
    int yearOfBirth =
Integer.parseInt(StringUtilities.daysToCalendarYear(visitStartDate)) -
row.getInt("AGE");
    person.add("year_of_birth", yearOfBirth);
    person.add("month_of_birth", "");
    person.add("day_of_birth", "");
} else if (row.getInt("AGEDAY") >= 0) {
    long dateOfBirth = visitStartDate - row.getInt("AGEDAY");
    person.add("year_of_birth", StringUtilities.daysToCalendarYear(dateOfBirth));
    person.add("month_of_birth", StringUtilities.daysToCalendarMonth(dateOfBirth));
    person.add("day_of_birth", StringUtilities.daysToCalendarDayOfMonth(dateOfBirth));
} else {
    person.add("year_of_birth", "");
    person.add("month_of_birth", "");
    person.add("day_of_birth", "");
}

person.add("race_source_value", row.get("RACE"));
if (row.get("RACE").equals("1")) // White
    person.add("race_concept_id", "8527");
else if (row.get("RACE").equals("2")) // Black
    person.add("race_concept_id", "8516");
else if (row.get("RACE").equals("4")) // Pacific islander
    person.add("race_concept_id", "8557");
else if (row.get("RACE").equals("5")) // Native American
    person.add("race_concept_id", "8657");
else if (row.get("RACE").equals("3")) // Hispanic, should be coded as 'other'
    person.add("race_concept_id", "8522");
else if (row.get("RACE").equals("6")) // Other
    person.add("race_concept_id", "8522");
else
    person.add("race_concept_id", "");

if (row.get("RACE").equals("3")) {// Hispanic
    person.add("ethnicity_source_value", "3");
    person.add("ethnicity_concept_id", "38003563");
} else {
    person.add("ethnicity_source_value", "");
    person.add("ethnicity_concept_id", "0");
}
```

Table name: visit_occurrence

Reading from core



Destination Field	Source Field	Logic	Comment
person_id	key		
visit_occurrence_id		Auto generate	
place_of_service_source_value			
place_of_service_concept_id		9201	

visit_start_date	year amonth aweekend	visit date = year + amonth + ? (first day of month that is correct weekday (weekday or weekend))
		if amonth < 1 then amonth = 1
		if aweekend < 0 then aweekend = 0	

visit_end_date	los	if los < 1 set to 0 (same day stay): start_date = end_date	There are some bogus values, but they are very infrequent.
care_site_id	hospid		

Sample code for computing visit start date

```
if (Integer.parseInt(amonth) < 1)
  amonth = Integer.toString(Math.abs(hash(key) % 12) + 1);
```

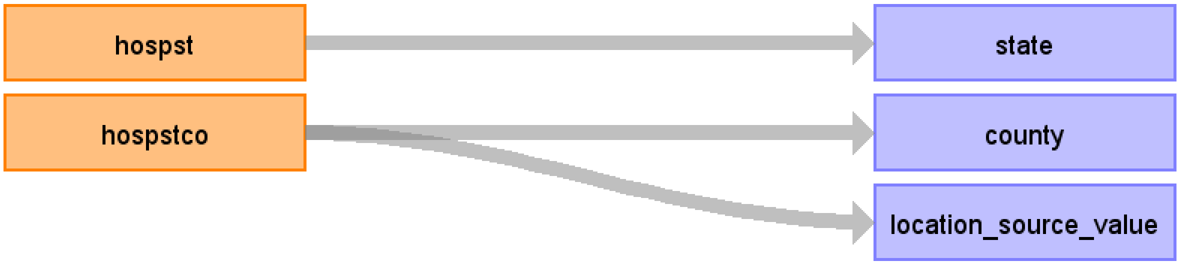
```
boolean isWeekend = aweekend.equals("1");
Calendar calendar = Calendar.getInstance();
calendar.set(Integer.parseInt(year), Integer.parseInt(amonth) - 1, 1);
while (isWeekend(calendar) != isWeekend)
  calendar.add(Calendar.DATE, 1);
long time = calendar.getTimeInMillis();
time += calendar.getTimeZone().getOffset(time);
// Millenium is added because for negative numbers, integer division truncates
upwards! (-8/10 = 0). This is reversed in the daysToDatabaseDateString function
visitStartDate = (((StringUtilities.MILLENIUM + time) / StringUtilities.DAY) - (1000 * 365));
```

Sample code for transformation:

```
Row visitOccurrence = new Row();
visitOccurrence.add("person_id", row.get("KEY"));
visitOccurrence.add("visit_occurrence_id", visitOccurrenceId);
visitOccurrence.add("visit_start_date",
StringUtilities.daysToDatabaseDateString(visitStartDate));
visitOccurrence.add("visit_end_date",
StringUtilities.daysToDatabaseDateString(visitEndDate));
visitOccurrence.add("care_site_id", row.get("HOSPID"));
visitOccurrence.add("place_of_service_concept_id", 9201); // Inpatient visit
```

Table name: location

Reading from core



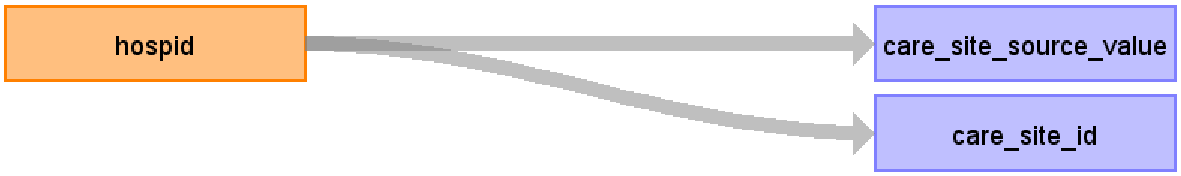
Destination Field	Source Field	Logic	Comment
location_id		For every combination of hospst + hospstco, a new location_id is generated. hoststco = NULL and hospstco = -9999 are considered to be the same.	
address_1			
address_2			
city			
state	hospst		
zip			
county	hospstco	Lookup in FIPS table http://www.census.gov/geo/reference/docs/codes/national_county.txt County name is truncated to 20 characters (note: the truncation has been verified to not create duplicate county names)	
location_source_value	hospstco		

Sample code for transformation:

```
String stateCounty = row.get("HOSPST") + "\t" + (row.get("HOSPSTCO").equals("-9999") ?
"" : row.get("HOSPSTCO"));
locationId = stateCountyToLocationId.get(stateCounty);
if (locationId == null) {
    locationId = stateCountyToLocationId.size() + 1;
    stateCountyToLocationId.put(stateCounty, locationId);

    Row location = new Row();
    location.add("location_id", locationId);
    location.add("state", row.get("HOSPST"));
    String county = codeToCounty.get(row.get("HOSPSTCO"));
    if (county == null)
        county = "";
    if (county.length() > 20)
        county = county.substring(0,20); //County field in CDM limited to 20 chars
    location.add("county", county);
    location.add("location_source_value", row.get("HOSPSTCO"));
}
```

Table name: care_site

Reading from core



Destination Field	Source Field	Logic	Comment
location_id		From location table based on hospst + hospstco	
organization_id			
place_of_service_concept_id			
place_of_service_source_value			
care_site_source_value	hospid		
care_site_id	hospid		

Sample code for transformation:

```
if (careSiteIds.add(row.getInt("HOSPID"))) {
    Row careSite = new Row();
    careSite.add("care_site_id", row.get("HOSPID"));
    careSite.add("care_site_source_value", row.get("HOSPID"));
    careSite.add("location_id", locationId);
}
```

Reading from hospital

Lots of nice data, but no place to put it in the CDM.

Table name: observation_period

Reading from core

Same logic as visit_occurrence table (visit = observation_period)

Destination Field	Source Field	Logic	Comment
observation_period_id			
person_id			
observation_period_start_date			
observation_period_end_date			

Sample code for transformation:

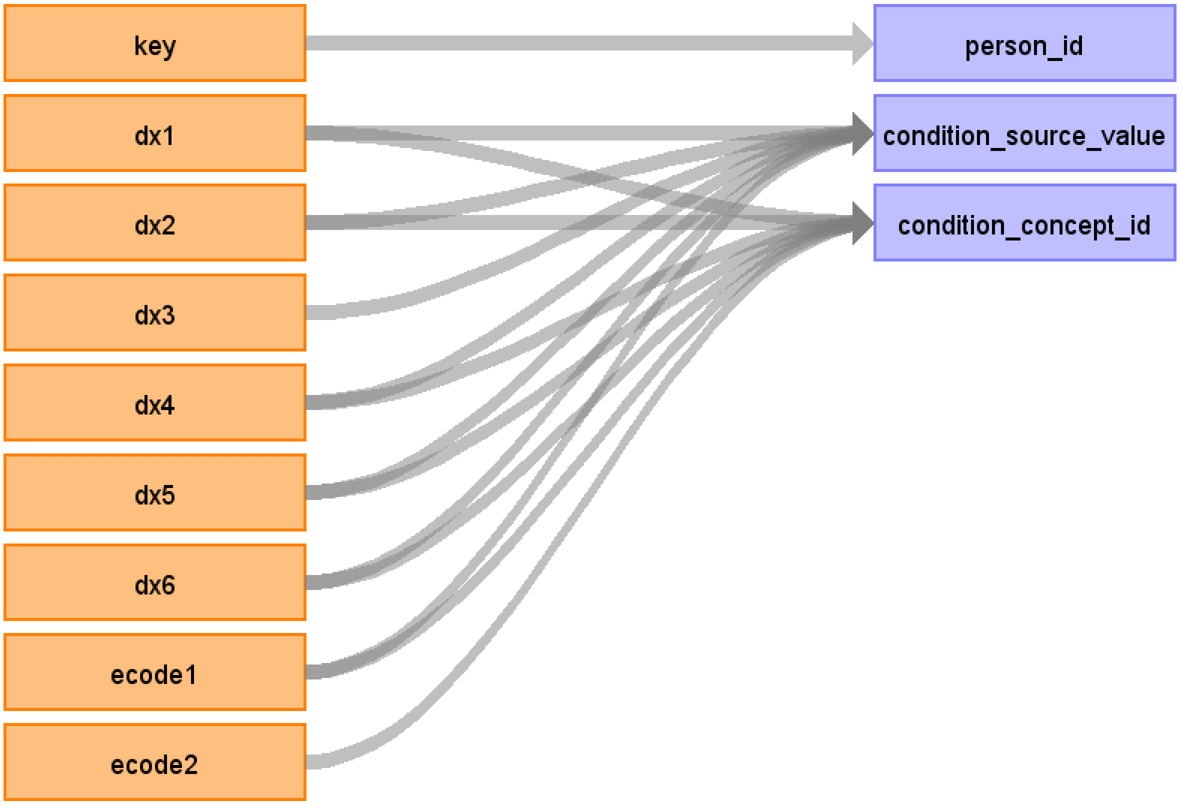
```
Row observationPeriod = new Row();
observationPeriod.add("observation_period_id", observationPeriodId);
observationPeriod.add("person_id", row.get("KEY"));
observationPeriod.add("observation_period_start_date",
StringUtilities.daysToDatabaseDateString(visitStartDate));
observationPeriod.add("observation_period_end_date",
StringUtilities.daysToDatabaseDateString(visitEndDate));
```

Table name: provider

Table name: condition_occurrence

Reading from core

If any of the diagnose fields maps to concept 4014295 (Single live birth) and the patient is either male or younger than 12, the code is discarted.



Destination Field	Source Field	Logic	Comment
person_id	key		
condition_occurrence_id			
condition_start_date	visit_start_date		
condition_end_date			
condition_type_concept_id	38000184 – 38000198 for dx1-dx15, respectively if ecode then 38000184-38000185		
stop_reason			

associated_provider_id			
visit_occurrence_id	From visit_occurrence table		
condition_source_value	dx1 dx2 dx3 dx4 dx5 dx6 ecode1		
condition_concept_id	dx1 dx2 dx4 dx5 dx6 ecode2 ecode1		

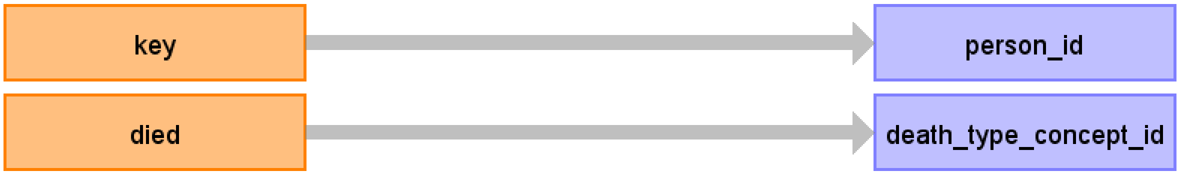
Sample code for transformation:

```
diagnoseFields = new String[] { "DX1", "DX2", "DX3", "DX4", "DX5", "DX6", "DX7",
"DX8", "DX9", "DX10", "DX11", "DX12", "DX13", "DX14", "DX15", "DX16", "DX17", "DX18",
"DX19", "DX20", "DX21", "DX22", "DX23", "DX24", "DX25", "ECODE1", "ECODE2" };
diagnoseFieldConceptIds = new int[] { 38000184, 38000185, 38000186, 38000187,
38000188, 38000189, 38000190, 38000191, 38000192, 38000193, 38000194, 38000195,
38000196, 38000197, 38000198, 38000198, 38000198, 38000198, 38000198, 38000198,
38000198, 38000198, 38000198, 38000198, 38000198, 38000184, 38000185
       };

for (int i = 0; i < diagnoseFields.length; i++)
    if (row.get(diagnoseFields[i]).trim().length() != 0) {
    if (conceptId == 4014295 && row.getInt("AGE") < 12) { // 4014295 = Single live
birth
      etlReport.reportProblem("Condition_occurrence", "Person < 12 years old with
live birth. Removing condition_occurrence", row.get("KEY"));
      continue;
    }
    if (conceptId == 4014295 && row.get("FEMALE").equals("0")) { // 4014295 = Single
live birth
      etlReport.reportProblem("Condition_occurrence", "Male with live birth. Removing
condition_occurrence", row.get("KEY"));
      continue;
     }
        Row conditionOccurrence = new Row();
        conditionOccurrence.add("person_id", row.get("KEY"));
        conditionOccurrence.add("condition_occurrence_id", ++conditionOccurrenceId);
        conditionOccurrence.add("condition_source_value", row.get(diagnoseFields[i]));
        conditionOccurrence.add("condition_concept_id", conceptId);
        conditionOccurrence.add("condition_type_concept_id",
diagnoseFieldConceptIds[i]);
        conditionOccurrence.add("condition_start_date",
StringUtilities.daysToDatabaseDateString(visitStartDate));
        conditionOccurrence.add("visit_occurrence_id", visitOccurrenceId);
        tableToRows.put("condition_occurrence", conditionOccurrence);
}
```

Table name: death

Reading from core

Only create a record if died = 1



Destination Field	Source Field	Logic	Comment
person_id	key		
death_date		visit_end_date	
death_type_concept_id		38003566 Medical claim discharge status "Died"	
cause_of_death_concept_id			
cause_of_death_source_value			

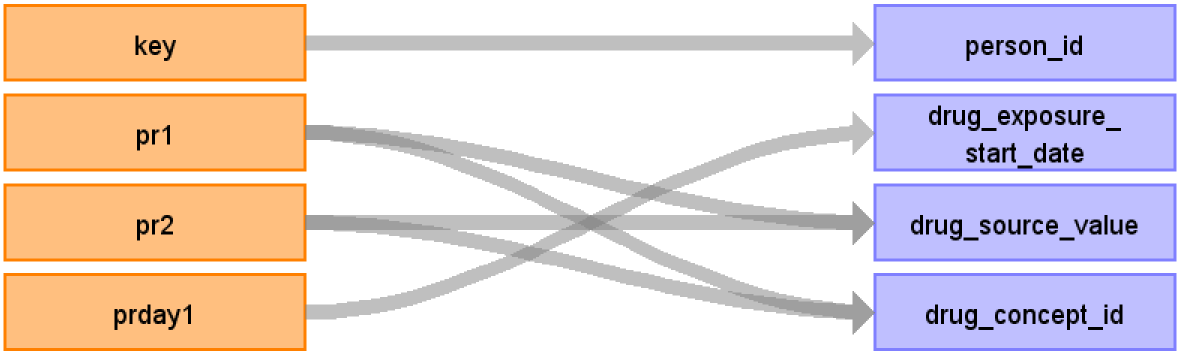
Sample code for transformation:

```
if (row.get("DIED").equals("1")) {
    Row death = new Row();
    death.add("person_id", row.get("KEY"));
    death.add("death_date", StringUtilities.daysToDatabaseDateString(visitEndDate));
    death.add("death_type_concept_id", 38003566); // EHR record patient status
"Deceased"
}
```

Table name: drug_exposure

Reading from core

A record is created only if procedure implies drug.



Destination Field	Source Field	Logic	Comment
person_id	key		
drug_exposure_start_date	prday1 prday2 …	If prdayx < 1 then visit_start_date, else (visit_start_date + prdayx)	
drug_exposure_end_date			
drug_type_concept_id		38000179 (Physician administered drug (identified as procedure))	
stop_reason			
refills			
quantity			
days_supply			
sig			
prescribing_provider_id			
visit_occurrence_id		From visit_occurrence table	
relevant_condition_concept_id			

drug_exposure_id		auto generate	
drug_source_value	pr1 pr2 …		
drug_concept_id	pr1 pr2 …	Lookup in mapping table	

Sample SQL code for finding procedures that map to drugs:

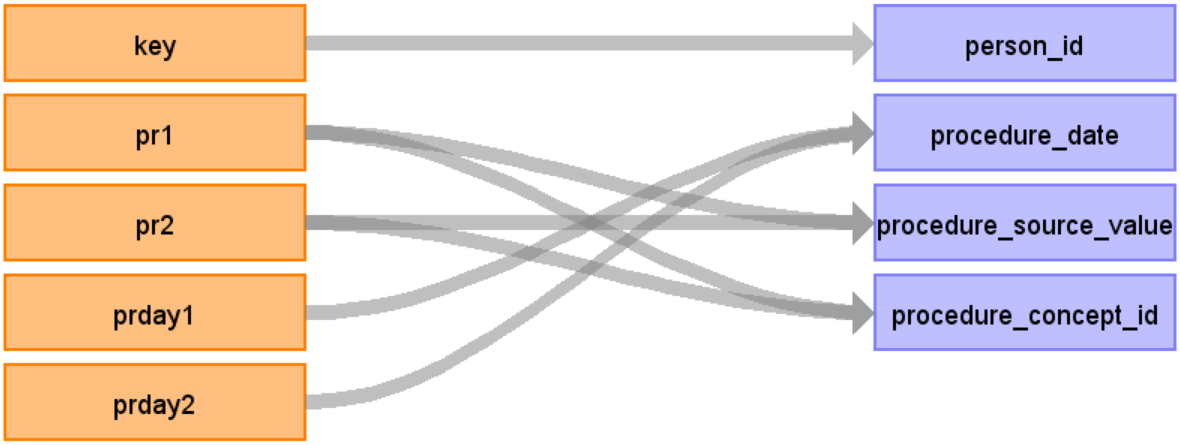
```
SELECT DISTINCT source_code, source_code_description, target_concept_id, concept_code,
concept_name FROM source_to_concept_map INNER JOIN concept ON target_concept_id = concept_id
WHERE  target_vocabulary_id = 8 AND source_vocabulary_id in (3,4,5) AND primary_map = 'Y' AND
COALESCE(source_to_concept_map.invalid_reason,’’) != 'D'
```

Sample code for transformation:

```
procedureFields = new String[] { "PR1", "PR2", "PR3", "PR4", "PR5", "PR6", "PR7",
"PR8", "PR9", "PR10", "PR11", "PR12", "PR13", "PR14", "PR15"};

procedureDayFields = new String[] { "PRDAY1", "PRDAY2", "PRDAY3", "PRDAY4", "PRDAY5",
"PRDAY6", "PRDAY7", "PRDAY8", "PRDAY9", "PRDAY10", "PRDAY11", "PRDAY12", "PRDAY13",
"PRDAY14", "PRDAY15" };

for (int i = 0; i < procedureFields.length; i++)
    if (row.get(procedureFields[i]).trim().length() != 0) {
        int conceptId = icd9ToRxNorm.getConceptId(row.get(procedureFields[i]).trim());
        if (conceptId != 0) {
            Row drugExposure = new Row();
            drugExposure.add("drug_exposure_id", ++drugExposureId);
            drugExposure.add("person_id", row.get("KEY"));
            int day = row.getInt(procedureDayFields[i]);
            if (day < 0)
                day = 0;
            if (day > visitEndDate - visitStartDate) {
               etlReport.reportProblem("Drug_exposure", "Drug exposure date beyond
length of stay, removing drug exposure", row.get("KEY"));
               continue;
            }
            drugExposure.add("drug_exposure_start_date",
StringUtilities.daysToDatabaseDateString(visitStartDate + day));
            drugExposure.add("drug_source_value", row.get(procedureFields[i]));
            drugExposure.add("drug_concept_id", conceptId);
            drugExposure.add("drug_type_concept_id", 38000179); // Physician
administered drug (identified as procedure)
            drugExposure.add("visit_occurrence_id", visitOccurrenceId);
            tableToRows.put("drug_exposure", drugExposure);
        }
    }
```

Table name: procedure_occurrence

Reading from core



Destination Field	Source Field	Logic	Comment
person_id	key		
procedure_occurrence_id		Auto generate	
procedure_type_concept_id		38000251 to 38000265, for pr1 through pr15	
procedure_date	prday1 prday2 …	If prdayx < 1 then visit_start_date, else (visit_start_date + prdayx)	
associated_provider_id			
visit_occurrence_id		from visit occurrence table	
relevant_condition_concept_id			
procedure_source_value	pr1 pr2 …		
procedure_concept_id	pr1 pr2 …	Lookup in source_to_concept_map	

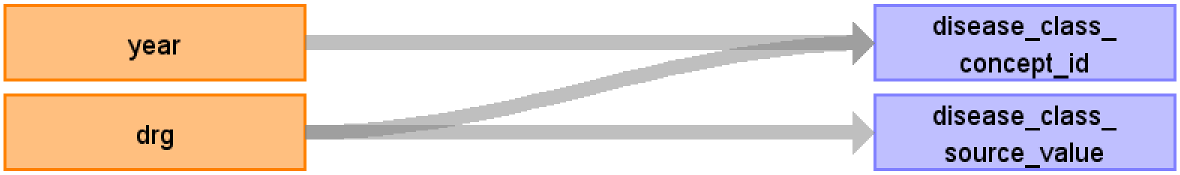
Sample code for transformation:

```
procedureFields = new String[] { "PR1", "PR2", "PR3", "PR4", "PR5", "PR6", "PR7",
"PR8", "PR9", "PR10", "PR11", "PR12", "PR13", "PR14", "PR15"};
```

```
procedureDayFields = new String[] { "PRDAY1", "PRDAY2", "PRDAY3", "PRDAY4", "PRDAY5",
"PRDAY6", "PRDAY7", "PRDAY8", "PRDAY9", "PRDAY10", "PRDAY11", "PRDAY12", "PRDAY13",
"PRDAY14", "PRDAY15" };

procedureFieldConceptIds = new int[] { 38000251, 38000252, 38000253, 38000254,
38000255, 38000256, 38000257, 38000258, 38000259, 38000260, 38000261, 38000262,
38000263, 38000264, 38000265 };


for (int i = 0; i < procedureFields.length; i++)
    if (row.get(procedureFields[i]).trim().length() != 0) {
        int day = row.getInt(procedureDayFields[i]);
        if (day < 0)
            day = 0;
        if (day > visitEndDate-visitStartDate){
          etlReport.reportProblem("Procedure_occurrence", "Procedure date beyond
length of stay, removing procedure", row.get("KEY"));
          continue;
        }
        Row procedureOccurrence = new Row();
        procedureOccurrence.add("procedure_occurrence_id", ++procedureOccurrenceId);
        procedureOccurrence.add("person_id", row.get("KEY"));
        procedureOccurrence.add("procedure_date",
StringUtilities.daysToDatabaseDateString(visitStartDate + day));
        procedureOccurrence.add("procedure_source_value",
row.get(procedureFields[i]));
        procedureOccurrence.add("procedure_concept_id",
icd9ProcToConcept.getConceptId(row.get(procedureFields[i]).trim()));
        procedureOccurrence.add("procedure_type_concept_id",
procedureFieldConceptIds[i]);
        procedureOccurrence.add("visit_occurrence_id", visitOccurrenceId);
        tableToRows.put("procedure_occurrence", procedureOccurrence);

        if (procedureFields[i].equals("PR1"))
        primaryProcedureOccurrenceId = procedureOccurrenceId;
    }
```

Table name: procedure_cost

Reading from core



Destination Field	Source Field	Logic	Comment
procedure_cost_id		Autogenerate, starts at value 1	
procedure_occurrence_id		procedure_occurrence_id corresponding to PR1	
paid_copay			
paid_coinsurance			
paid_toward_deductible			
paid_by_payer			
paid_by_coordination_benefits			
total_out_of_pocket			
total_paid			
revenue_code_concept_id			
payer_plan_period_id			
revenue_code_source_value			
disease_class_concept_id	drg year	Use year to select only the DRG code in the vocab that was valid that year	
disease_class_source_value	drg		

Sample code for transformation:

```
if (primaryProcedureOccurrenceId != -1) {
    String drg = row.get("DRG").trim();
    int conceptId;
    if (StringUtilities.isInteger(drg)) {
        conceptId = drgYearToConcept.getConceptId(String.format("%03d",
Integer.parseInt(drg)) + "_" + row.get("YEAR"));
    } else
        conceptId = drgYearToConcept.getConceptId(drg + "_" + row.get("YEAR"));

    procedureCostId++;
    Row procedureCost = new Row();
    procedureCost.add("procedure_cost_id", procedureCostId);
    procedureCost.add("procedure_occurrence_id", primaryProcedureOccurrenceId);
    procedureCost.add("disease_class_source_value", row.get("DRG"));
    procedureCost.add("disease_class_concept_id", conceptId);
}
```

Table name: observation

Table name: condition_era

Table name: drug_era

Table name: organization

Table name: payer_plan_period

Table name: drug_cost

Table name: cohort

Appendix: source tables

Table: core

Field	Type	Most freq. value	Comment
key	Integer	List truncated...	
female	Integer	1	
year	Integer	2006	
amonth	Integer	-9	
aweekend	Integer	0	
age	Integer	0	
ageday	Integer	-99	
race	Integer	1	
hospst	VarChar	CA	
hospstco	Integer		Indicates the five (or four) digits of zip code for hospital
hospid	Integer	51043	
los	Integer	2	
los_x	Integer	2	
asource	Integer	5	
asource_x	VarChar		
atype	Integer	1	
died	Integer	0	
discwt	Real	4.6286	
dispub92	Integer		
dispuniform	Integer	1	
dqtr	Integer	1	

drg	Integer	391	DRG
drgver	Integer	22	
drg18	Integer		DRG 18
drg24	Integer		DRG 24
dshospid	VarChar		
dx1	VarChar	V3000	
dx2	VarChar		
dx3	VarChar		
dx4	VarChar		
dx5	VarChar		
dx6	VarChar		
dx7	VarChar		
dx8	VarChar		
dx9	VarChar		
dx10	Free text	4019	
dx11	Free text	4019	
dx12	VarChar		
dx13	VarChar		
dx14	VarChar		
dx15	VarChar		
dxccs1	Integer	218	CCS = clinical classification systems
dxccs2	Integer	-999	
dxccs3	Integer	-999	
dxccs4	Integer	-999	
dxccs5	Integer	-999	

dxccs6	Integer	-999	
dxccs7	Integer	-999	
dxccs8	Integer	-999	
dxccs9	Integer	-999	
dxccs10	Integer	-999	
dxccs11	Integer	-999	
dxccs12	Integer	-999	
dxccs13	Integer	-999	
dxccs14	Integer	-999	
dxccs15	Integer	-999	
mdc	Integer	5	
mdc18	Integer		
mdnum1_s	VarChar		
mdnum2_s	VarChar		
ndx	Integer	9	
neomat	Integer	0	
nis_stratum	Integer	3033	
npr	Integer	0	
pay1	Integer	1	
pay1_x	VarChar	C	
pay2	Integer	-9	
pay2_x	VarChar		
pr1	VarChar		
pr2	VarChar		
pr3	VarChar		

pr4	VarChar		
pr5	VarChar		
pr6	Free text	8853	
pr7	VarChar		
pr8	VarChar		
pr9	VarChar		
pr10	VarChar		
pr11	VarChar		
pr12	VarChar		
pr13	VarChar		
pr14	VarChar		
pr15	Integer		
prccs1	Integer	-99	
prccs2	Integer	-99	
prccs3	Integer	-99	
prccs4	Integer	-99	
prccs5	Integer	-99	
prccs6	Integer	-99	
prccs7	Integer	-99	
prccs8	Integer	-99	
prccs9	Integer	-99	
prccs10	Integer	-99	
prccs11	Integer	-99	
prccs12	Integer	-99	
prccs13	Integer	-99	

prccs14	Integer	-99	
prccs15	Integer	-99	
prday1	Integer	-99	
prday2	Integer	-99	
prday3	Integer	-99	
prday4	Integer	-99	
prday5	Integer	-99	
prday6	Integer	-99	
prday7	Integer	-99	
prday8	Integer	-99	
prday9	Integer	-99	
prday10	Integer	-99	
prday11	Integer	-99	
prday12	Integer	-99	
prday13	Integer	-99	
prday14	Integer	-99	
prday15	Integer	-99	
totchg	Integer	-999999999	
totchg_x	Real	-99999999999.99	
zipinc	Integer		
discwtcharge	Real		
mdid_s	VarChar		
surgid_s	VarChar		
asourceub92	VarChar		
dispub04	Integer		

dqtr_x	Integer		
drg_nopoa	Integer		
dx16	VarChar		
dx17	VarChar		
dx18	VarChar		
dx19	VarChar		
dx20	VarChar		
dx21	VarChar		
dx22	VarChar		
dx23	VarChar		
dx24	VarChar		
dx25	VarChar		
dxccs16	Integer		
dxccs17	Integer		
dxccs18	Integer		
dxccs19	Integer		
dxccs20	Integer		
dxccs21	Integer		
dxccs22	Integer		
dxccs23	Integer		
dxccs24	Integer		
dxccs25	Integer		
ecode1	VarChar		
ecode2	VarChar		
ecode3	VarChar		

ecode4	VarChar		
elective	Integer	0	
e_ccs1	Integer	-999	
e_ccs2	Integer	-999	
e_ccs3	Integer	-999	
e_ccs4	Integer	-999	
hcup_ed	Integer		
hospbrth	Integer		
mdc24	Integer		
mdc_nopoa	Integer		
nchronic	Integer		
necode	Integer	0	
orproc	Integer		
pl_nchs2006	Integer		
pointoforiginub04	VarChar		
pointoforigin_x	VarChar		
tran_in	Integer		
tran_out	Integer		
zipinc_qrtl	Integer		
mdnum1_r	Integer		
mdnum2_r	Integer		
pl_ur_cat4	Integer		

Table: dx_pr_grps

Field	Type	Most freq. value	Comment
hospid	Integer	12320	

key	Integer	List truncated...	
chron1	Integer	0	
chron2	Integer	0	
chron3	Integer	1	
chron4	Integer	1	
chron5	Integer	1	
chron6	Integer	-99	
chron7	Integer	-99	
chron8	Integer	-99	
chron9	Integer	-99	
chron10	Integer	-99	
chron11	Integer	-99	
chron12	Integer	-99	
chron13	Integer	-99	
chron14	Integer	-99	
chron15	Integer	-99	
chron16	Integer		
chron17	Integer		
chron18	Integer		
chron19	Integer		
chron20	Integer		
chron21	Integer		
chron22	Integer		
chron23	Integer		
chron24	Integer		

chron25	Integer		
chronb1	Integer	7	
chronb2	Integer	7	
chronb3	Integer	7	
chronb4	Integer	-99	
chronb5	Integer	-99	
chronb6	Integer	-99	
chronb7	Integer	-99	
chronb8	Integer	-99	
chronb9	Integer	-99	
chronb10	Integer	-99	
chronb11	Integer	-99	
chronb12	Integer	-99	
chronb13	Integer	-99	
chronb14	Integer	-99	
chronb15	Integer	-99	
chronb16	Integer		
chronb17	Integer		
chronb18	Integer		
chronb19	Integer		
chronb20	Integer		
chronb21	Integer		
chronb22	Integer		
chronb23	Integer		
chronb24	Integer		

chronb25	Integer		
dxmccs1	VarChar		
e_mccs1	VarChar		
pclass1	Integer	-99	
pclass2	Integer	-99	
pclass3	Integer	-99	
pclass4	Integer	-99	
pclass5	Integer	-99	
pclass6	Integer	-99	
pclass7	Integer	-99	
pclass8	Integer	-99	
pclass9	Integer	-99	
pclass10	Integer	-99	
pclass11	Integer	-99	
pclass12	Integer	-99	
pclass13	Integer	-99	
pclass14	Integer	-99	
pclass15	Integer	-99	
prmccs1	VarChar		
ccsmgn1	Integer		
ccsmgn2	Integer		
ccsmgn3	Integer		
ccsmgn4	Integer		
ccsmgn5	Integer		
ccsmgn6	Integer		

ccsmgn7	Integer		
ccsmgn8	Integer		
ccsmgn9	Integer		
ccsmgn10	Integer		
ccsmgn11	Integer		
ccsmgn12	Integer		
ccsmgn13	Integer		
ccsmgn14	Integer		
ccsmgn15	Integer		
ccsmsp1	Integer		
ccsmsp2	Integer		
ccsmsp3	Integer		
ccsmsp4	Integer		
ccsmsp5	Integer		
ccsmsp6	Integer		
ccsmsp7	Integer		
ccsmsp8	Integer		
ccsmsp9	Integer		
ccsmsp10	Integer		
ccsmsp11	Integer		
ccsmsp12	Integer		
ccsmsp13	Integer		
ccsmsp14	Integer		
ccsmsp15	Integer		
eccsmgn1	Integer		

eccsmgn2	Integer		
eccsmgn3	Integer		
eccsmgn4	Integer		

Table: hospital

Field	Type	Most freq. value	Comment
ahaid	VarChar		
discwt	Real	4.4322	
hospaddr	VarChar		
hospcity	VarChar		
hospid	Integer	List truncated...	
hospname	VarChar		
hospst	VarChar	TX	
hospwt	Real	5.0000	
hospzip	Integer		
hosp_bedsize	Integer	1	
hosp_control	Integer	0	
hosp_location	Integer	1	
hosp_locteach	Integer	2	
hosp_region	Integer	3	
hosp_teach	Integer	0	
idnumber	VarChar		
nis_stratum	Integer	2411	
n_disc_u	Integer	133443	
n_hosp_u	Integer	78	
s_disc_u	Integer	84786	

s_hosp_u	Integer	16	
total_disc	Integer	List truncated...	
year	Integer	2008	
discwtcharge	Real		
hfipsstco	Integer		
h_contrl	Integer		
hospstco	Integer	-9999	
hosp_rnpct	Integer		
hosp_rnfteapd	Real		
hosp_lpnfteapd	Real		
hosp_nafteapd	Real		
hosp_opsurgpct	Integer		
hosp_mhsmember	Integer		
hosp_mhscluster	Integer		

Table: severity

Field	Type	Most freq. value	Comment
hospid	Integer	51043	
key	Integer	List truncated...	
aprdrg	Integer	640	
aprdrg_risk_mortality	Integer	1	
aprdrg_severity	Integer	1	
cm_aids	Integer	0	
cm_alcohol	Integer	0	
cm_anemdef	Integer	0	
cm_arth	Integer	0	

cm_bldloss	Integer	0	
cm_chf	Integer	0	
cm_chrnlung	Integer	0	
cm_coag	Integer	0	
cm_depress	Integer	0	
cm_dm	Integer	0	
cm_dmcx	Integer	0	
cm_drug	Integer	0	
cm_htn_c	Integer	0	
cm_hypothy	Integer	0	
cm_liver	Integer	0	
cm_lymph	Integer	0	
cm_lytes	Integer	0	
cm_mets	Integer	0	
cm_neuro	Integer	0	
cm_obese	Integer	0	
cm_para	Integer	0	
cm_perivasc	Integer	0	
cm_psych	Integer	0	
cm_pulmcirc	Integer	0	
cm_renlfail	Integer	0	
cm_tumor	Integer	0	
cm_ulcer	Integer	0	
cm_valve	Integer	0	
cm_wghtloss	Integer	0	

ds_dx_category1	VarChar		
ds_stage1	Real	1.01	
apsdrg	Integer		
apsdrg_charge_weight	Real		
apsdrg_los_weight	Real		
apsdrg_mortality_weight	Real		
ds_los_level	Integer		
ds_los_scale	Real		
ds_mrt_level	Integer		
ds_mrt_scale	Real		
ds_rd_level	Integer		
ds_rd_scale	Real		