Questions about MSDW2 OMOP Data from a Collaborator

Questions Submitted on 11/03/2022

Background

The following questions came up while working with the MSDW2 data in Version 1.6.220610 of June 14th. Some of these might already have been addressed in more recent versions.

Scientific Computing Reply:

Since June 14, when we provided a collaborator with a “snapshot” extract of our OMOP tables, our MSDW2 development has focused mostly on bug fixes and infrastructure improvements, rather than adding new types of data records to our OMOP tables. (Notable exception: We have added flowsheet records to MEASUREMENT (v1.9) and OBSERVATION (v1.10) on October 7 and October 19, respectively.)

Consequently, the collaborator’s June “snapshot” extract is very much outdated. MSDW2’s architecture supports full change data capture (CDC) on all tables, so it would be relatively straightforward to establish an incremental extraction process (i.e., only those records that have changed since your last extraction). Such an incremental extraction process would enable the collaborator to keep the OMOP copies updated more easily than periodic bulk snapshots.

Questions

1. Some conditions/findings seem to be exclusively in i2b2 and/or MSDW2

We tried to find out in a systematic way if and what codes are not present in MSDW2, but it was very difficult due to ambiguous mapping of ICD - EPIC - SNOMED. For a specific cohort, we compared the

number of unique patientID+Date tuples (not looking into specific diagnosis codes due to different underlying terminologies) and found that this differs between i2b2 and MSDW2 data. Maybe we could go over specific examples to try to find out where the respective diagnosis might be stored in the MSDW2 data schema?

Scientific Computing Reply:

We do not recommend attempting to compare the data in MSDW2 with any of its predecessor databases such as the former MSDW (on Oracle), the former OMOP (on Postgres), and the former i2b2 (on Postgres). The data sources, quality of the mappings, length of history covered, architecture, and data pipelines were drastically different from those of MSDW2 today. The deficiencies of these prior databases were driving factors in the conception, design, and business case for replacing them all with MSDW2.

We took great pains to establish transparent data lineage from Epic to MSDW2, in terms of both structure (i.e., source-to-target mapping of database objects) and content (identifiers, codes, and categories created in Epic's Chronicles database). The source-to-target mapping information is scanned and available in the Informatica Enterprise Data Catalog (EDC) being implemented by Jeff Quinn's team. MSDW2's use of Epic's identifiers is pervasive throughout the OMOP data model, but perhaps isn't obvious. This is one area where we could improve our documentation.

Unique identifiers from Epic's Chronicles database may be found in the following locations in MSDW2's OMOP database:

OMOP Database Column	**Description**
Extension columns with the prefix "`xtn_epic`"	When we mapped a source Caboodle table to a target OMOP table, we included any additional columns from that Caboodle table that we judged could be of value to researchers. These extension columns include the Epic Chronicles unique identifiers for important transactions or master files, like patients, providers, departments, encounters, orders, etc.
ETL-related columns with the prefix "`etl_epic`"	These columns support transparent data lineage for our data pipeline. Most of these columns have a "`_key`" or "`_durable_key`" suffix, which match the source tables in Caboodle.
`concept.concept_code` on records where `vocabulary_id` begins with "EPIC"	We load Epic's master files and category lists to the CONCEPT table for use throughout the OMOP database in columns with the suffix "`_source_concept_id`". Because OMOP's `concept.concept_id` column is a surrogate primary key, assigned by our OMOP load process to avoid value "collisions" across multiple vocabularies, we store Epic's original "business key" identifiers in the CONCEPT table's `concept_code` column.

OMOP Database Column	Description
Columns with names like "*<name>*_source_value"	OHDSI intends for these optional OMOP columns to hold the text descriptions that are mapped to concepts for their corresponding *<name>*_concept_id columns, whether or not the *<name>*_source_concept_id columns are populated. In our OMOP database, we always populate the *<name>*_source_concept_id columns. Therefore, the value of every *<name>*_source_value column will match the concept.concept_name column for its corresponding source concept record.

The former i2b2 database was loaded from our legacy MSDW database (on Oracle). The legacy MSDW was loaded from HL7 messages copied from Mount Sinai's Cloverleaf integration engine, from data extracts from Epic Clarity, and from a subset of Mount Sinai's billing data extracted as flat files from the EAGLE hospital biling system. While there should be a high degree of overlap between diagnosis codes used for billing/revenue cycle and diagnosis codes used for clinical documentation in Epic, we know that this overlap is always imperfect. Integrating Mount Sinai's billing data from the MSX data mart into MSDW2's OMOP tables is an item on our roadmap "wish list".

2. EPIC codes are given as source values of the CONDITION_OCCURENCE table. Would it be possible to also get information on ICD source values that were put into the system for a corresponding SNOMED ID?

Scientific Computing Reply:

Yes, this information is already available in the CONCEPT_RELATIONSHIP table. See details below.

Unfortunately, OHDSI has declared that SNOMED-CT is the standard vocabulary for CONDITION_OCCURRENCE, despite the fact that almost all academic researchers in the U.S. use ICD-10-CM for conditions and diagnoses. We share OMOP-formatted data with enough external research partners that we did not feel justified in breaking interoperability by deviating from this OHDSI "rule": the values populated in condition_concept_id should be standard, according to OHDSI. Such concepts have concept.standard_concept = 'S'.

In additional to the OMOP-compliant mappings of Epic diagnoses to standard SNOMED-CT codes, we also established our own mappings of Epic diagnoses to ICD-10-CM codes. To do so, we defined a custom relationship_id in the RELATIONSHIP table for use in CONCEPT_RELATIONSHIP. We have engaged an external vendor, Clinical Architecture LLC, to map our seven major Epic master files to healthcare vocabularies, including the mapping of Epic diagnoses to ICD-10-CM codes. In full disclosure, we have not yet applied Clinical Architecture's mappings to the data in our OMOP tables (as of November 14, 2022). But we have loaded the mappings that exist in Epic from the diagnosis master file to ICD-10-CM codes (which may exhibit a one-to-many relationship). See the following query:

```
SELECT
 c1.vocabulary_id
,c1.concept_code
,c1.concept_name
,cr.relationship_id
,c2.vocabulary_id
,c2.concept_code
,c2.concept_name
FROM cdm_phi.concept_relationship cr
JOIN cdm_phi.concept c1
  ON cr.concept_id_1 = c1.concept_id
JOIN cdm_phi.concept c2
  ON cr.concept_id_2 = c2.concept_id
WHERE cr.relationship_id = 'Maps to non-standard'
  AND   c1.vocabulary_id = 'EPIC EDG .1'
  AND   c2.vocabulary_id = 'ICD10CM'
ORDER BY
 c1.concept_code
,c2.concept_code
```

You can utilize our “non-standard” mappings from Epic diagnoses to ICD-10-CM and ICD-9-CM codes to identify patients in the CONDITION_OCCURRENCE table using queries like the following:

```
SELECT
 COUNT(DISTINCT dx.person_id) AS pt_count
,COUNT(*)                     AS dx_count
FROM cdm_phi.condition_occurrence dx
JOIN (
   SELECT DISTINCT
   c1.concept_id
    FROM cdm_phi.concept_relationship cr
    JOIN cdm_phi.concept c1
      ON cr.concept_id_1  = c1.concept_id
        AND c1.vocabulary_id = 'EPIC EDG .1'
    JOIN cdm_phi.concept c2
      ON cr.concept_id_2  = c2.concept_id
   WHERE cr.relationship_id = 'Maps to non-standard'
     AND ((c2.vocabulary_id = 'ICD10CM' AND c2.concept_code BETWEEN 'E08' AND 'E13')
          OR (c2.vocabulary_id = 'ICD9CM'  AND c2.concept_code LIKE '250%')
         )
    ) epic
ON dx.condition_source_concept_id = epic.concept_id
```

3. PERSON_IDs without entry in PERSON table

These were potentially addressed in data update from Oct 15? (Screenshot of release notes below)

Version 1.9.221015 – October 15, 2022

Fixed	• In some cases, the records in OMOP tables pertaining to patients switching from regulatory non-protected status to protected status were not getting updated to match. The OMOP person table did not exhibit this problem.

Scientific Computing Reply:

We have made several bug fixes that improve the referential integrity of foreign-key relationships between OMOP tables. One such bug fix was release 1.9.221015 on October 15, as mentioned above. Another such bug fix was release 1.8.220915 on September 15.

The Epic EHR has complex data processing to handle the merging of patient records when a clinician discovers that one person in real life has more than one medical record in Epic. The Epic EHR system also has the ability to "unmerge" a patient record if it was erroneously merged. These patient record merges and unmerges are not handled very gracefully downstream in Clarity and Caboodle. We are continuing to monitor these rare cases and make refinements to our data pipeline to avoid having "duplicate" patient records.

On a somewhat related topic, we are currently (as of November 14, 2022) working to fix a known bug where systolic and diastolic blood pressure measurements in the MEASUREMENT table are sometimes linked to the wrong VISIT_OCCURRENCE record.

4. Note title (clinical note type) often missing for clinical notes - is this information not available or could this potentially be included in the notes table?

Scientific Computing Reply:

For clinical notes loaded to the NOTE table, the column `note_title` and extension column `xtn_note_class_source_concept_id` are populated using data element INP 5010 from Epic's Chronicles database. This data element INP 5010 is a category list, which we have not yet mapped to standard OMOP concepts, as the following query demonstrates. Our mapping of these note types (or note "classes" as they are called in OMOP) is underway.

```
SELECT
 c.concept_id
,c.vocabulary_id
,c.concept_code
,c.concept_name
,(CASE WHEN cr.concept_id_1 IS NOT NULL
       THEN 1 ELSE 0 END) AS has_mapping
FROM cdm_phi.concept c
LEFT JOIN cdm_phi.concept_relationship cr
ON c.concept_id = cr.concept_id_1
AND cr.relationship_id = 'Maps to'
WHERE c.vocabulary_id = 'EPIC INP 5010'
ORDER BY c.concept_code
```

Some laboratory tests and procedures have lengthy text reports, which we also load to the NOTE table (under a different `xtn_note_type_source_concept_id`). There is no obvious source data element for the `note_title` and `note_class_concept_id` columns for these records, but we are continuing to explore possibilities.

We are actively working to add radiology reports to the NOTE table as well. These reports are composed of multiple free-text fields in Epic – separate sections for narrative, impressions, and any addenda -- so we plan to load each section as a separate row in NOTE.

5. 8% of the observation_dates are on 2015-10-23, is there a reason for that (e.g. change of a system)?

Scientific Computing Reply:

Yes, these dates are an artifact of a system migration in which a large batch of patient records were loaded into Epic from a legacy system.

We load patient demographics data elements into the OBSERVATION table. Because these are "header" data elements, there isn't a good date to use for the OBSERVATION table's required `observation_date` column. Consequently, we use the creation date of each patient's record in Epic.

Please see the query below for the demographics data elements pertaining to the batch of patient records created on October 23, 2015:

```
SELECT
 o.observation_concept_id
,o.observation_concept_name
,o.observation_source_concept_id
,o.observation_source_concept_name
,count(*)
FROM cdm_phi.observation o
WHERE o.xtn_observation_type_source_concept_name = 'Patient Demographics'
AND o.observation_date = CAST('2015-10-23' AS DATE)
GROUP BY
 o.observation_concept_id
,o.observation_concept_name
,o.observation_source_concept_id
,o.observation_source_concept_name
ORDER BY
 o.observation_concept_name
```

6. Missing mappings ("metadata")

Recently, for instance, we searched for specific medication prescriptions (Upadacitinib (Rinvoq); Ozanimod (Zeposia); Risankizumab (Skyrizi)) and realized that these should rather be searched for as strings in the DRUG_SOURCE_VALUE column as mappings for most of these prescribed medications are not yet established. Would you recommend in general to always double check with source value columns and string searches while mappings are not complete yet?

Scientific Computing Reply:

Yes. Because our mappings from Epic's medications to standard RxNorm codes is not complete, the approach you suggest is similar to the one we use when writing queries to extract data for researchers' custom data requests.

Specifically, we construct a value set of Epic medication identifiers for the medications of interest, then apply that value set as a filter on the DRUG_EXPOSURE table. You have choices for how to apply this value-set filter, because there are two pairs of columns each loaded with the same Epic data element:

Epic Data Element	OMOP Columns

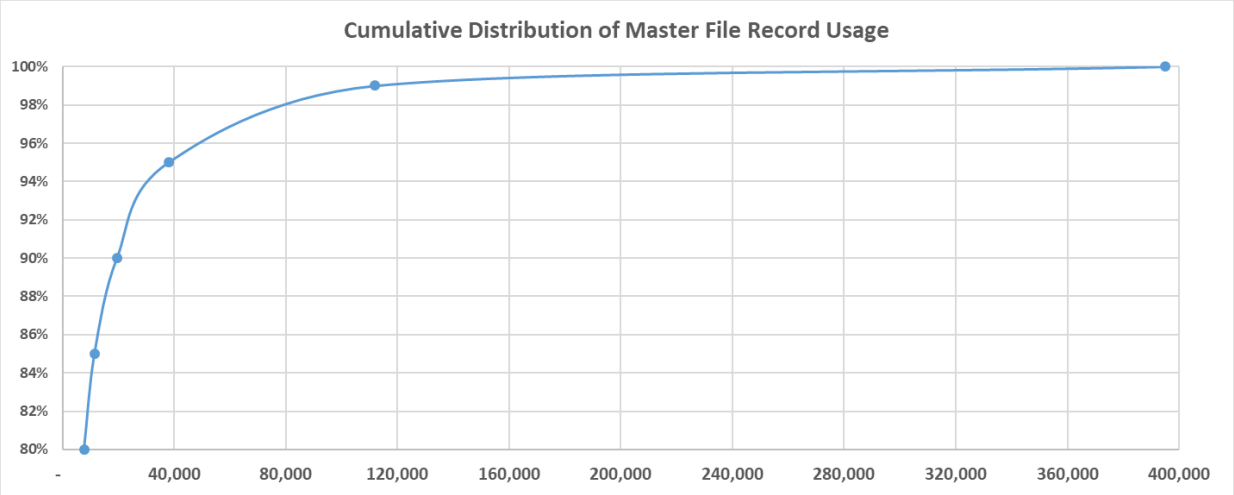
ERX .1	`drug_exposure.xtn_epic_medication_id`
	
	`concept.concept_code`
	(where `vocabulary_id = 'EPIC ERX .1'`)

ERX .2	`drug_exposure.drug_source_value`
	
	`concept.concept_name`
	(where `vocabulary_id = 'EPIC ERX .1'`)

Because the CONCEPT table only includes the single attribute `concept_name`, we added the other medication attributes to the DRUG_EXPOSURE table as extension columns. For the three medications listed in your example, you could write a query such as the following:

```
SELECT DISTINCT
 d.drug_source_concept_id
,d.xtn_epic_medication_id
,d.drug_source_concept_code
,d.drug_source_concept_name
,d.drug_source_value
,d.xtn_drug_generic_name
,d.xtn_generic_ingredient_source_concept_name
FROM cdm_phi.drug_exposure d
WHERE d.xtn_drug_generic_name LIKE '%upadacitinib%'
   OR d.xtn_drug_generic_name LIKE '%ozanimod%'
   OR d.xtn_drug_generic_name LIKE '%risankizumab%'
```

The usage of master file records in Epic exhibit a consistent pattern: a few codes account for the majority of all transactions, and there is a very long "tail" of infrequently used codes. See diagram and table below:



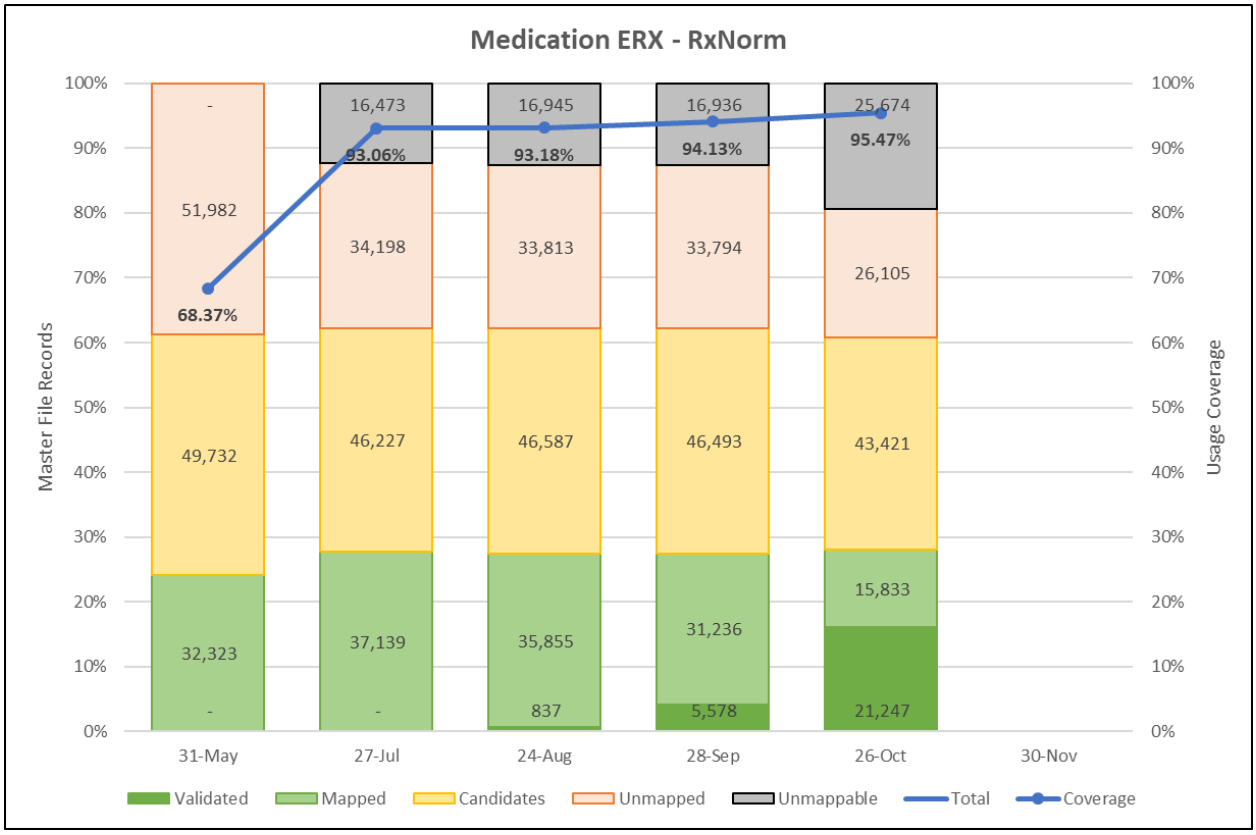
Priority	Master File	Target Vocabulary	Rows for 80%	Rows for 85%	Rows for 90%	Rows for 95%	Rows for 99%	Used Rows	Total Rows
1	Diagnosis EDG	SNOMED-CT	6,000	9,000	16,000	32,000	96,000	324,721	1,566,543
2	Diagnosis EDG	ICD10CM	6,000	9,000	16,000	32,000	96,000	324,721	1,566,543
3	Lab Component LRR	LOINC / SNOMED	200	300	400	600	1,800	21,866	33,891
4	Medication ERX	RxNorm	800	1,000	1,400	2,400	6,200	53,376	134,037
5	Procedure EAP	CPT / HCPCS	200	400	600	1,000	3,000	26,159	154,086
6	Surgical Procedure ORP	CPT / HCPCS	400	600	800	1,400	2,600	4,712	31,486
7	Immunization LIM	CVX / RxNorm	30	40	50	80	130	333	399
8	Allergy ELG	SNOMED-CT / RxNorm	150	200	350	700	2,300	8,249	38,914
		TOTAL:	**7,780**	**11,540**	**19,600**	**38,180**	**112,030**	**439,416**	**1,959,356**
			80%	85%	90%	95%	99%	100%	

As of November 14, 2022, we have mapped 7,213 medications, which account for 87% of all transactions in the DRUG_EXPOSURE table.

```
SELECT
(CASE WHEN c2.concept_id IS NOT NULL THEN 'Mapped' ELSE 'Unmapped' END) AS is_mapped
,COUNT(*) AS medication_count
FROM cdm_phi.concept c1
LEFT JOIN (
   cdm_phi.concept_relationship cr
   JOIN cdm_phi.concept c2
     ON cr.concept_id_2  = c2.concept_id
       AND c2.vocabulary_id = 'RxNorm'
)
 ON c1.concept_id      = cr.concept_id_1
AND cr.relationship_id = 'Maps to'
WHERE c1.vocabulary_id = 'EPIC ERX .1'
GROUP BY
(CASE WHEN c2.concept_id IS NOT NULL THEN 'Mapped' ELSE 'Unmapped' END)
```

```
SELECT
 cast(sum(case when d.drug_source_concept_id != 0 then 1 else 0 end) * 100.0 /
      count_big(*) as numeric(18,3)) AS Source_Fill_Pct
,cast(sum(case when d.drug_concept_id        != 0 then 1 else 0 end) * 100.0 /
      sum(case when d.drug_source_concept_id != 0 then 1 else 0 end) as numeric(18,3))
         AS Mapped_Pct
from cdm_phi.drug_exposure d
where d.drug_exposure_id != 0
```

The progess by Clinical Architecture LLC on mapping our Epic medications master file, as mentioned in Question #2 above, as of October 26, 2022 is depicted in the graph below. As mentioned above, we have not yet applied these mappings from Clinical Architecture to our OMOP database, but we hope to do so in the near future.



7. Out-of-domain entries

Is there a reason why entries from different domains are present in the individual tables (screenshot below summarizing domains of mapped entries from data version Jun 14 Version 1.6.220610)? What would be a recommendation to best work with the data? (e.g. extracting data from same domain from different tables?)

Table	Domain	Frequency
CONDITION_OCCURRENCE	Condition	80.21 %
	Observation	16.05 %
	Metadata	2.64 %
	Procedure	0.54 %
	Measurement	0.54 %
	Meas Value	0.02 %
DRUG_EXPOSURE	Drug	86.75 %
	Metadata	13.25 %
PROCEDURE_OCCURRENCE	Measurement	56.36 %
	Procedure	22.51 %
	Metadata	18.89 %
	Drug	1.17 %
	Observation	1.06 %
	Device	0.01 %
MEASUREMENT	Measurement	82.34 %
	Metadata	14.54 %
	Observation	3.12 %
OBSERVATION	Observation	61.52 %
	Condition	14.84 %
	Procedure	12.14 %
	Measurement	6.29 %
	Unit	3.13 %
	Race	2.08 %
	Metadata	0.0 %

Scientific Computing Reply:

OHDSI's "rule" for standardizing content within the OMOP model is to map source values to "standard" concepts that align with the domain of that column. For example, all concepts populating the column `condition_occurrence.condition_concept_id` should have `concept.standard_concept = 'S'` and `concept.domain_id = 'Condition'`.

Unfortunately, the records within Epic's master files and their usage within the EHR system do not strictly adhere to this "rule". Ideally, Epic's diagnosis master file would contain only true diagnoses, Epic's procedure master file would contain only true procedures, and so on, but this is not the case in practice. There are many master file records that contain "miscellaneous" concepts for convenient clinical documentation within particular contexts or workflows within the Epic EHR system.

Because of this situation, we included the following "Unresolved" item in our release notes since our first 1.0 release on September 20, 2021. We would like to move all such "out-of-domain" Epic records to the OBSERVATION table, but this work is not very high on our priority list.

Unresolved	• Epic's documentation records that appear in the diagnosis and procedure master files aren't yet re-assigned to OMOP's observation table (from condition_occurrence and procedure_occurrence, respectively).

Some of our source Epic values are mapped to concepts in seemingly incongruous OMOP domains. It is possible that some of these mappings are suboptimal choices produced by the fuzzy matching algorithm

in OHDSI's Usagi software tool. We expect that Clinical Architecture's mapping work (mentioned in Question #2 above) will greatly improve the quality of our concept mappings.

Regarding the use of concept domains within the OBSERVATION table, you should be aware of the following guidance from OHDSI (source: https://ohdsi.github.io/CommonDataModel/cdm53.html#OBSERVATION):

> **User Guide**
>
> Observations differ from Measurements in that they do not require a standardized test or some other activity to generate clinical fact. Typical observations are medical history, family history, the stated need for certain treatment, social circumstances, lifestyle choices, healthcare utilization patterns, etc. If the generation clinical facts requires a standardized testing such as lab testing or imaging and leads to a standardized result, the data item is recorded in the MEASUREMENT table. If the clinical fact observed determines a sign, symptom, diagnosis of a disease or other medical condition, it is recorded in the CONDITION_OCCURRENCE table. **Valid Observation Concepts are not enforced to be from any domain though they still should be Standard Concepts.** [*emphasis added*]

We believe it is safe to disregard the "out-of-domain" concepts in each OMOP table, because these are unlikely to be proper drugs, procedures, lab test results, etc. Additionally, our source-to-target mapping of Epic's database tables to our OMOP tables is correct. For example, there is no risk of a medication administration or a performed procedure accidentally showing up in the OBSERVATION table (or elsewhere). Epic's database tables are tied to particular clinical workflows in the Epic EHR system.

When in doubt, we recommend the query process outlined in our answer to Question #6 above.

8. Most entries in the measurements table lack valid information on units of values

Scientific Computing Reply:

The MEASUREMENT table is loaded from several different source tables in Epic, so any meaningful analysis of the units of measure should be done by source table. Here are our mappings by source, as of November 14, 2022:

```
SELECT
 m.xtn_measurement_type_source_concept_id
,m.xtn_measurement_type_source_concept_name
,(CASE WHEN m.unit_concept_id != 0 THEN 'Has units' ELSE 'No units' END) AS has_units
,count(*)
FROM cdm_phi.measurement m
GROUP BY
 m.xtn_measurement_type_source_concept_id
,m.xtn_measurement_type_source_concept_name
,(CASE WHEN m.unit_concept_id != 0 THEN 'Has units' ELSE 'No units' END)
ORDER BY
 m.xtn_measurement_type_source_concept_id
,(CASE WHEN m.unit_concept_id != 0 THEN 'Has units' ELSE 'No units' END)
```

xtn_measurement_type_source_concept	Has Units	No Units	Total	Percent
Lab Component Result	330,146,586	371,521,507	701,668,093	47.1%
Vital Signs	325,741,788	117,241,412	442,983,200	73.5%
Flowsheet Measurement	109,890,678	437,891	110,328,569	99.6%

Unfortunately, the units of measure data element for lab components (ORD 2050 in the Epic Chronicles database) is not a category list in Epic. Therefore, we construct a "source vocabulary" for OMOP's CONCEPT table by performing a SELECT DISTINCT operation across all extant values in Epic's transaction table. As a result, there is a larger-than-expected volume of such values, and their data quality is much lower, than would be the case if this data element were a controlled category list. To date, we have only established mappings for only 83/1689 = 5% of these values.

```
SELECT
(CASE WHEN c2.concept_id IS NOT NULL THEN 'Mapped' ELSE 'Unmapped' END) AS is_mapped
,COUNT(*) AS units_count
FROM cdm_phi.concept c1
LEFT JOIN (
   cdm_phi.concept_relationship cr
   JOIN cdm_phi.concept c2
     ON cr.concept_id_2  = c2.concept_id
       AND c2.vocabulary_id = 'UCUM'
)
 ON c1.concept_id      = cr.concept_id_1
AND cr.relationship_id = 'Maps to'
WHERE c1.vocabulary_id = 'EPIC ORD 2050'
GROUP BY
(CASE WHEN c2.concept_id IS NOT NULL THEN 'Mapped' ELSE 'Unmapped' END)
```