Cooking with CQL Qs&As – Session 40

Thursday, December 5, 2019

Measure Logic In CQL

Q: CMS826v0 - Hospital Harm - Pressure Injury – How would you express an encounter with a new harmful pressure injury?

A: Paraphrased description of the measure: This measure assesses the proportion of inpatient encounters of patients 18 years of age or older at admission, who have a pressure ulcer present on admission (POA) and suffer the harm of developing a new pressure injury (stage 2, stage 3, stage 4, deep tissue pressure injury, or nonstageable pressure injury) subsequent to admission. The measure does not consider the number of new pressure injuries per encounter, it is only concerned about a new pressure injury in patients admitted with an existing pressure injury. Thus, the measure must identify a pressure injury that is POA and a new pressure injury in a different anatomical location identified at least 24 hours after arrival. This new harmful pressure injury's prevalence period overlaps, but starts after an existing pressure ulcer.

For this particular encounter, the measure is looking for the presence of an Encounter diagnosis consistent with codes in a specific value set 'diagnosis: Pressure Ulcer Stage' with a 'Pressure injury stage' (e.g., containing values: stage 2, stage 3, stage 4, nonstageable pressure ulcer, pressure injury of deep tissue). This approach will set up an alias for pressure injury POA where that diagnosis (harmful pressure injury) has a POA indicator. Next, we are looking for a 'harmful pressure injury' during the hospitalization period where the anatomical location site is not the same as the one that was POA and it starts more than 24 hours after the start of hospitalization.

Option 1 (using 'let'):

```
define "Encounter With New Harmful Pressure Injury":
      "Denominator" EncounterWithQualifyingAge
               let PressureInjuryPOA:
                      "Harmful Pressure Injury" PressureInjury
                             where PressureInjury.prevalencePeriod
                                    during "Hospitalization, Potentially Starting in
      Emergency Department and or with Observation"(EncounterWithQualifyingAge)
                                    and exists (
                                           EncounterWithQualifyingAge.diagnoses
      EncounterDiagnosis
                                                  where EncounterDiagnosis.code ~
      PressureInjury.code // Best approximation in QDM, no way to link encounter diagnoses
      directly to Diagnosis elements
```

```
                                                    and
      EncounterDiagnosis.presentOnAdmissionIndicator ~ "y"
                                        )
                  where exists ("Harmful Pressure Injury" PressureInjury
                    where PressureInjury.prevalencePeriod during "Hospitalization,
      Potentially Starting in Emergency Department and or with
      Observation"(EncounterWithQualifyingAge)
                            and AllTrue(
                                        PressureInjuryPOA POA
                                          where PressureInjury.prevalencePeriod starts
      more than 24 hours after start of "Hospitalization, Potentially Starting in Emergency
      Department and or with Observation"(EncounterWithQualifyingAge)
                                                    and
      PressureInjury.anatomicalLocationSite !~ POA.anatomicalLocationSite
                                                    return true
                                    )
                  )
```

Option 2 (using a 'with' statement in place of the 'let' and include a 'such that' statement, only if there is one condition):

```
define "Encounter With New Harmful Pressure Injury":
              "Denominator" EncounterWithQualifyingAge
                     with (
                            "Harmful Pressure Injury" PressureInjury
                                   where PressureInjury.prevalencePeriod
                                          during "Hospitalization, Potentially Starting in
              Emergency Department and or with Observation"(EncounterWithQualifyingAge)
                                                 and exists (
                                   EncounterWithQualifyingAge.diagnoses EncounterDiagnosis
                                                                 where
              EncounterDiagnosis.code ~ PressureInjury.code // Best approximation in QDM, no
              way to link encounter diagnoses directly to Diagnosis elements
                                                                 and
              EncounterDiagnosis.presentOnAdmissionIndicator ~ "y"
                                                                 )
                                          )  PressureInjuryPOA
                                                 such that exists ("Harmful Pressure
              Injury" PressureInjury
                                                    where PressureInjury.prevalencePeriod
              during "Hospitalization, Potentially Starting in Emergency Department and or
              with Observation"(EncounterWithQualifyingAge)
```

```
                                                    and AllTrue(
                                                                PressureInjuryPOA
POA
                                                                  where
PressureInjury.prevalencePeriod starts more than 24 hours after start of
"Hospitalization, Potentially Starting in Emergency Department and or with
Observation"(EncounterWithQualifyingAge)
                                                                        and
PressureInjury.anatomicalLocationSite !~ POA.anatomicalLocationSite
                                                                    return true
                                                        )
                                                )
```

The difference between the 'with' and 'let' is that the 'let' introduces "PressureInjuryPOA" so you can talk about it in multiple places throughout the query, whereas the 'with' only lets you talk about it within 'such that'. If you need to talk about it in a subsequent 'with' or 'where' then the 'let' permits you to repeatedly talk about the same expression. The 'with' only references the Pressure Injury POA within the 'such that' condition. The scope of the 'let' variable goes on after the initial 'where' or filtering that is done. As far as performance, expressing it as a 'with' or a 'let' depends on how the underlying implementation decides to deal with it because it might translate into a 'join' in a structured query language database environment vs. an 'apply'. It is up to the optimizer of the target system to determine how best to run it.

Regardless of whether the expression uses 'let' or 'with', the following issue may represent a challenge in testing and evaluating this measure. A potential challenge with this approach is the terminology used when recording an anatomical site. For example, if the anatomical location site is recorded as hip ulcer, but didn't specify laterality (right or left) and then a new hip ulcer developed on the other hip, the straight code comparison here wouldn't differentiate the two. Conversely, if the existing ulcer code indicates "greater trochanter," and a separate reference to the same pressure ulcer used the code "hip," the result might suggest there are two pressure ulcers when only one exists. Thus coding variation could lead to missing a new pressure ulcer or over-counting an existing pressure ulcer as newly developed. A possible solution would be to develop very specific value sets that distinguish the encounter.

Q: For the measure CMS646, Bladder Cancer (https://oncprojectracking.healthit.gov/support/browse/CQLIT-206), how do you express specific references for bladder cancer and combine several definitions into one?

A: The clinical workflow presented by the measure developer is that bladder cancer is diagnosed and then the tumor is staged and coded with either Ta HG (noninvasive papillary carcinoma – high grade), Tis (carcinoma in situ), or T1 (tumor has spread to the connective tissue). Bacillus Calmette-Guerin (BCG) is the standard immunotherapy drug (regardless of tumor stage) for bladder cancer and will be administered up to six times, but for this measure we are only interested in the first administration. The intent of this measure is to assure that staging occurs prior to the first BCG administration. The measure only looks at the first BCG administration,

not at BCG dose/frequency/etc., even if the staging occurred up to six months prior; we are interested in the first encounter after the bladder cancer diagnosis within the measurement period. Note, the diagnosis must overlap the measurement period. The first BCG given is associated with the staging and obviously the encounter. We are specifically checking that the BCG administered happened after the staging.

The suggested approach is to look for any qualifying encounter that occurred during the submission period and we define Qualifying Encounter, Initial Population, Bladder Cancer Staging, Bladder Cancer Diagnosis, and First BCG Administered as:

Qualifying Encounter:

```
define "Qualifying Encounter":
                ["Encounter, Performed": "Office Visit"] Encounter
                      where Encounter.relevantPeriod during "Measurement Period"
```

Initial Population:

```
define "Initial Population:
      exists ( "Bladder Cancer Diagnosis" )
        and "Most Recent Bladder Cancer Staging Tumor Ta HG, Tis, T1"
        and exists "Qualifying Encounter"
```

Bladder Cancer Staging:

```
Define "Most Recent Bladder Cancer Staging Tumor Ta HG, Tis, T1":
     "Bladder Cancer Staging".result ~ "T1: Tumor invades lamina propria or
   submucosa (finding)"
       or "Bladder Cancer Staging".result ~ "Ta: Noninvasive papillary carcinoma
   (urinary tract) (finding)"
       or "Bladder Cancer Staging".result in ( "Bladder Cancer Staging Tis for
   Urology Care" )
```

Bladder Cancer Diagnosis:

```
Define "Bladder Cancer Diagnosis":
               ["Diagnosis": "Bladder Cancer for Urology Care"] BladderCancer
                 where BladderCancer.prevalencePeriod starts same day or before
                 end "Measurement Period"
```

First BCG Administered:

```
define "First BCG Administered":
             First(["Medication, Administered": "BCG Bacillus Calmette Guerin for Urology
          Care"] BCG
                with "Bladder Cancer Diagnosis" BladderCancer
                  such that BCG.relevantPeriod starts 6 months or less after start
          BladderCancer.prevalencePeriod
                    and BCG.relevantPeriod overlaps "Measurement Period"
                sort by start of relevantPeriod
            )
```

Solutions:

1. Use exists() function to provide Boolean (true/false) results for Initial Population definitions.
2. Remove singleton from office visit expression since timing requirements addressed by other definitions.
3. Use ~ equivalence function for code but use in() function to Value Set Authority Center object identifiers for tumor staging definition.

Q: For measure CMS111, Median Admit Decision Time to ED Departure Time for Admitted Patients (https://oncprojectracking.healthit.gov/support/browse/CQLIT-207), we are having a measure observation calculation issue, where some duration calculation results are based on the ending interval (end time of encounter). If the emergency department (ED) encounter does not include the facility location, it should not calculate a "Visit" encounter for the measure observation time. If the ED facility location departure time is missing, a measure observation time should not be calculated, but in Bonnie this does occur. Is this due to the logic below? Should we add a null clause in the Initial Population to indicate if there is no ED facility location departure time the encounter will not be included?

```
Global.HospitalizationLocations (Encounter "Encounter, Performed") Encounter Visit
    let EDVisit: Last(["Encounter, Performed": "Emergency Department Visit"] LastED
        where LastED.relevantPeriod ends 1 hour or less on or before start of
          Visit.relevantPeriod
        sort by end of relevantPeriod)
        return if EDVisit is null then Visit.facilityLocations
            else flatten
            { EDVisit.facilityLocations, Visit.facilityLocations }
```

A: The measure intent is to capture the median time interval from when the admit decision is made (represented by an AssessmentPerformed.relevantDatetime or by an AdmitOrder.authorDatetime) to when the patient leaves the ED (represented by the end of HospitalLocation.locationPeriod).

```
define function "AdmitDecisionUsingAssessment"(Encounter "Encounter, Performed" ):
    Last(["Assessment, Performed": "ED Evaluation"] EDEvaluation
          where EDEvaluation.relevantDatetime during
    "RelatedEDVisit"(Encounter).relevantPeriod
            and EDEvaluation.result in "Admit Inpatient"
          sort by relevantDatetime
      )
```

The “EDDepartureTime”(“RelatedEDVisit”(Encounter)) is represented by the end of HospitalLocations.locationPeriod. However, if HospitalLocations.code is present (ED Department), but the end of the locationPeriod is missing (null), then the time interval calculation would continue to infinity. If the ED Location EndTime was not noted, the measure performs the calculation until infinity, or the end of time, instead of to zero. To address this calculation issue, the initial encounter is expressed so that it includes an EDEncounter code and could not be null for the EDDepartureTime. In this case, the only ones passing through the measure would have an observation measure time.

```
define “Initial Population”:
    Global."Inpatient Encounter" EncounterInpatient
        with "ED Encounter with Decision to Admit" EDAdmit
            such that EDAdmit.relevantPeriod 1 hour or less before or on start of
EncounterInpatient.relevantPeriod


define “ED Encounter with Decision to Admit”:

  ( ( Global."ED Encounter" EDVisit where
"AdmitDecisionUsingAssessmentDuringRelatedEDBeforeDeparture"(EDVisit).relevantDatetime

        during EDVisit.relevantPeriod)

    union ( Global."ED Encounter" EDVisit where
"AdmitDecisionUsingEncounterOrderDuringRelatedEDandBeforeDeparture"(EDVisit).authorDatetime

        during EDVisit.relevantPeriod

  ) ) EDVisitAdmit

  where EDVisitAdmit.facilityLocations.locationPeriod is not null
```

If an ED visit is followed by an inpatient encounter, it’s safe to assume there’s missing data that wasn’t reported so you could assert that the end of the location period where it isn’t specified is the start of the inpatient encounter that wasn’t specified. Another thing you could do is to introduce a stratification that would detect the cases where EDFacilityLocation doesn’t have an ending date. This is a way to get the measure to indicate how many we’re getting that don’t have an end date documented. The measure observation would still be calculated but the AdmitTime would take place of the EDDepatureTime.

Using Quality Data Model (QDM)

Q: In QDM 5.5, we have reduced ability to talk about entity types and these are attributes that are patients, persons, or practitioners. In the examples we’ve used to illustrate these is using the ‘is’ operator in CQL by saying participant is an organization. However, there is an issue with Bonnie that it doesn’t support the ‘is’ operator, but it does support the ‘as’ operator. How do we work around this issue?
A: In the example below, it is using the ‘as’ operator to get to the same place by turning that into Org and saying it’s not null. Here we’re saying participant ‘as’ an organization. If the participant is not an organization, then the result of that is null so by saying that if Org1 is not null and Org2 is not null, then you can do the comparison between them.

```
define "My Qualifying Encounters":
              ( distinct ( flatten ( from
                      "Dementia Encounter" Enc1,
                      "Dementia Encounter" Enc2
                      let Org1: ( Enc1.participant as "Organization" ),
                      Org2: ( Enc2.participant as "Organization" )
```

```
            where Org1 is not null
              and Org2 is not null
              and start of Enc2.relevantPeriod 1 day or more after start
of Enc1.relevantPeriod
              and Org1.identifier ~ Org2.identifier
            return all { Enc1, Enc2 }
        )
      )
    )
```