Cooking with CQL Q&As

Session 59 - Thursday, December 9, 2021

# Measure Logic in CQL

**Q:** Is the data model used to create the MedicationRequestPeriod function in the cumulative medication duration Fast Healthcare Interoperability Resource (FHIR®) 4.0.1 CQL library part of US Core profile?

**A:**  Yes, the model used is based on the US Core 4.0.0 profile, which is based on FHIR® 4.0.1 specification. Information regarding the US Core Implementation Guide is located at URL hl7.org/fhir/us/core

**Q:** The MedicationRequestPeriod function in the cumulative medication duration CQL library is using Fast Healthcare Interoperability Resource (FHIR®) 4.0.1 but not US Core 4.0.0, is this correct?

**A:** Correct. There is still some further investigation and tooling development needed to provide the essential capabilities to reference US Core, QI Core, and other implementation guides directly. Eventually you would have the ability to change the expression from FHIR® to US Core, QI Core, or whatever implementation guide is appropriate.

**Q:** When calculating the Cumulative Medication Duration using Medication Dispense data, are the functions all executed using the Atom plug-in for CQL? Do you tell the Atom plug-in what version of CQL or execution engine to run?

**A:** Yes, the Cumulative Medication Duration is calculated with the Atom plug-in for CQL. The Atom plug-in downloads a translator to support the language, compiling, and error messages as well as an evaluator to support running the test. There is also the same Atom plug-in for Visual Studio Code if that is your preferred environment.

**Q:** In the Cumulative Medication Duration expression (collapse Interval per day), does the per day need to be declared?

|  |  |  |
| --- | --- | --- |
| define function CumulativeDuration(Intervals List<Interval<Date>>): | | |
|  | Sum((collapse Intervals per day) X return all difference in days between start of X and end of X) + 1 |

**A:** The per day interval needs to be declared or it will collapse at the least granular interval in the expression. For example, if you have intervals per minute, it will collapse the interval to the minute so specifying per day is important. Otherwise, the logic will consider dates on sequential days as gaps. For example, if the event ends at 10 AM on Saturday and starts at 10 AM on Sunday, from a day perspective, those meet, equal one day and would collapse. If the calculation were performed per minute, there would be a gap of 24 hours.

# Using Quality Data Model

**Q:** In the QDM (Quality Data Model) library related to medication dispense period, is a relevant date and time allowed to be used as a starting point when other data points are absent?

**A:** The Dispense.relevantDatetime attribute means the date and time the prescription dispensing event occurred (the whenHandedOver if mapped to FHIR). Dispense.relevantPeriod is the time period for which the dispensed supply is to be administered/taken, not including refills. The QDM data type allows for either to be specified, but not both. So, either an instance of a Dispense gives the period over which it is to be taken, or it gives a date and time the event occurred. Because the logic is looking for a starting point, it will take whichever element is specified as the starting point.

If Dispense.relevantdatetime is specified, it may be used as a starting point for the medication dispense period. The order of the attributes within the Coalesce operator expression determines the relative priority of Dispense.relevantdatetime, Dispense.relevantperiod.low, and Dispense.authorDatetime data elements.

Any given instance should not have both a Dispense.relevantdatetime and a Dispense.relevantperiod.low, so whichever dispense information is present will be given priority. If neither Dispense.relevantdatetime or Dispense.relevantperiod.low is available, then the Dispense.authorDatetime will be used to calculate the medication dispense startDate.

Similarly, totalDaysSupplied uses a Coalesce of Dispense.daysSupplied if available, otherwise calculates it from the Dispense.supply.value divided by the Dispense.dosage.value times the frequency. For example, Rx of 1 tab twice a day #60, totalDaysSupplied = 60 tabs /(1 tab/dose x 2 dose/day) = 30 day supply

|  |  |
| --- | --- |
| define function "MedicationDispensedPeriod"(Dispense "Medication, Dispensed" ): | |
|  | Dispense Dispense |
|  | let |
|  | startDate: date from Coalesce(Dispense.relevantDatetime, Dispense.relevantPeriod.low, Dispense.authorDatetime), |
|  | totalDaysSupplied: |
|  | Coalesce( |
|  | Dispense.daysSupplied, |
|  | Dispense.supply.value / (Dispense.dosage.value \* ToDaily(Dispense.frequency)) |
|  | ) |