CQLIT-266 Expand Intervals Part A 20210319

Created test environment in MAT/Bonnie to evaluate your examples and more:

//Integer, Closed LO, Closed HI, (default "per 1", including "per 1" causes MAT error)   
define "Question1x":  
 expand { Interval[1, 5]}  
[INTERVAL: 1 - 1, INTERVAL: 2 - 2, INTERVAL: 3 - 3, INTERVAL: 4 - 4, INTERVAL: 5 – 5]

//Integer, Closed LO, Open HI  
define "Question2x":  
 expand { Interval[1, 5) }  
[INTERVAL: 1 - 1, INTERVAL: 2 - 2, INTERVAL: 3 - 3, INTERVAL: 4 - 4]

//Decimal, Closed LO, Closed HI  
define "Question3x":  
 expand { Interval[1.0, 5.0]}  
[INTERVAL: 1 - 1, INTERVAL: 2 - 2, INTERVAL: 3 - 3, INTERVAL: 4 - 4, INTERVAL: 5 - 5]

define "Question3x2":  
 expand { Interval[1.0, 5.0]} per 2.0  
[INTERVAL: 1 - 2, INTERVAL: 3 - 4]

define "Question3x3":  
 expand { Interval[1.0, 5.0]} per 0.1  
[INTERVAL: 1 - 1.09999999, INTERVAL: 1.1 - 1.19999999, INTERVAL: 1.2 - 1.29999999, INTERVAL: 1.3 - 1.39999999, INTERVAL: 1.4 - 1.49999999, INTERVAL: 1.5 - 1.59999999, INTERVAL: 1.6 - 1.69999999, INTERVAL: 1.7 - 1.79999999, INTERVAL: 1.8 - 1.89999999, INTERVAL: 1.9 - 1.99999999, INTERVAL: 2 - 2.09999999, INTERVAL: 2.1 - 2.19999999, INTERVAL: 2.2 - 2.29999999, INTERVAL: 2.3 - 2.39999999, INTERVAL: 2.4 - 2.49999999, INTERVAL: 2.5 - 2.59999999, INTERVAL: 2.6 - 2.69999999, INTERVAL: 2.7 - 2.79999999, INTERVAL: 2.8 - 2.89999999, INTERVAL: 2.9 - 2.99999999, INTERVAL: 3 - 3.09999999, INTERVAL: 3.1 - 3.19999999, INTERVAL: 3.2 - 3.29999999, INTERVAL: 3.3 - 3.39999999, INTERVAL: 3.4 - 3.49999999, INTERVAL: 3.5 - 3.59999999, INTERVAL: 3.6 - 3.69999999, INTERVAL: 3.7 - 3.79999999, INTERVAL: 3.8 - 3.89999999, INTERVAL: 3.9 - 3.99999999, INTERVAL: 4 - 4.09999999, INTERVAL: 4.1 - 4.19999999, INTERVAL: 4.2 - 4.29999999, INTERVAL: 4.3 - 4.39999999, INTERVAL: 4.4 - 4.49999999, INTERVAL: 4.5 - 4.59999999, INTERVAL: 4.6 - 4.69999999, INTERVAL: 4.7 - 4.79999999, INTERVAL: 4.8 - 4.89999999, INTERVAL: 4.9 - 4.99999999]

//Decimal, Closed LO, Open HI  
define "Question4x":  
 expand { Interval[1.0, 5.0) }  
[INTERVAL: 1 - 1, INTERVAL: 2 - 2, INTERVAL: 3 - 3, INTERVAL: 4 - 4]

define "Question4x2":  
 expand { Interval[1.0, 5.0) } per 2.0  
[INTERVAL: 1 - 2, INTERVAL: 3 – 4]

define "Question4x3":  
 expand { Interval[1.0, 5.0 ) } per 0.1  
[INTERVAL: 1 - 1.09999999, INTERVAL: 1.1 - 1.19999999, INTERVAL: 1.2 - 1.29999999, INTERVAL: 1.3 - 1.39999999, INTERVAL: 1.4 - 1.49999999, INTERVAL: 1.5 - 1.59999999, INTERVAL: 1.6 - 1.69999999, INTERVAL: 1.7 - 1.79999999, INTERVAL: 1.8 - 1.89999999, INTERVAL: 1.9 - 1.99999999, INTERVAL: 2 - 2.09999999, INTERVAL: 2.1 - 2.19999999, INTERVAL: 2.2 - 2.29999999, INTERVAL: 2.3 - 2.39999999, INTERVAL: 2.4 - 2.49999999, INTERVAL: 2.5 - 2.59999999, INTERVAL: 2.6 - 2.69999999, INTERVAL: 2.7 - 2.79999999, INTERVAL: 2.8 - 2.89999999, INTERVAL: 2.9 - 2.99999999, INTERVAL: 3 - 3.09999999, INTERVAL: 3.1 - 3.19999999, INTERVAL: 3.2 - 3.29999999, INTERVAL: 3.3 - 3.39999999, INTERVAL: 3.4 - 3.49999999, INTERVAL: 3.5 - 3.59999999, INTERVAL: 3.6 - 3.69999999, INTERVAL: 3.7 - 3.79999999, INTERVAL: 3.8 - 3.89999999, INTERVAL: 3.9 – 3.99999999]  
//Note: does not capture 4.x, might expect to 4.9 – 4.99999999 but loses whole unit

define "Question4x4":  
 expand { Interval[1.00, 5.00 ) } per 0.1  
[INTERVAL: 1 - 1.09999999, INTERVAL: 1.1 - 1.19999999, INTERVAL: 1.2 - 1.29999999, INTERVAL: 1.3 - 1.39999999, INTERVAL: 1.4 - 1.49999999, INTERVAL: 1.5 - 1.59999999, INTERVAL: 1.6 - 1.69999999, INTERVAL: 1.7 - 1.79999999, INTERVAL: 1.8 - 1.89999999, INTERVAL: 1.9 - 1.99999999, INTERVAL: 2 - 2.09999999, INTERVAL: 2.1 - 2.19999999, INTERVAL: 2.2 - 2.29999999, INTERVAL: 2.3 - 2.39999999, INTERVAL: 2.4 - 2.49999999, INTERVAL: 2.5 - 2.59999999, INTERVAL: 2.6 - 2.69999999, INTERVAL: 2.7 - 2.79999999, INTERVAL: 2.8 - 2.89999999, INTERVAL: 2.9 - 2.99999999, INTERVAL: 3 - 3.09999999, INTERVAL: 3.1 - 3.19999999, INTERVAL: 3.2 - 3.29999999, INTERVAL: 3.3 - 3.39999999, INTERVAL: 3.4 - 3.49999999, INTERVAL: 3.5 - 3.59999999, INTERVAL: 3.6 - 3.69999999, INTERVAL: 3.7 - 3.79999999, INTERVAL: 3.8 - 3.89999999, INTERVAL: 3.9 – 3.99999999]  
//Note: does not capture 4.x, might expect to 4.9 – 4.99999999 but loses whole unit

define "Question4x5":  
 expand { Interval[1.50, 5.50 ) } per 0.1  
[INTERVAL: 1.5 - 1.59999999, INTERVAL: 1.6 - 1.69999999, INTERVAL: 1.7 - 1.79999999, INTERVAL: 1.8 - 1.89999999, INTERVAL: 1.9 - 1.99999999, INTERVAL: 2 - 2.09999999, INTERVAL: 2.1 - 2.19999999, INTERVAL: 2.2 - 2.29999999, INTERVAL: 2.3 - 2.39999999, INTERVAL: 2.4 - 2.49999999, INTERVAL: 2.5 - 2.59999999, INTERVAL: 2.6 - 2.69999999, INTERVAL: 2.7 - 2.79999999, INTERVAL: 2.8 - 2.89999999, INTERVAL: 2.9 - 2.99999999, INTERVAL: 3 - 3.09999999, INTERVAL: 3.1 - 3.19999999, INTERVAL: 3.2 - 3.29999999, INTERVAL: 3.3 - 3.39999999, INTERVAL: 3.4 - 3.49999999, INTERVAL: 3.5 - 3.59999999, INTERVAL: 3.6 - 3.69999999, INTERVAL: 3.7 - 3.79999999, INTERVAL: 3.8 - 3.89999999, INTERVAL: 3.9 - 3.99999999, INTERVAL: 4 - 4.09999999, INTERVAL: 4.1 - 4.19999999, INTERVAL: 4.2 - 4.29999999, INTERVAL: 4.3 - 4.39999999, INTERVAL: 4.4 - 4.49999999, INTERVAL: 4.5 - 4.59999999, INTERVAL: 4.6 - 4.69999999, INTERVAL: 4.7 - 4.79999999, INTERVAL: 4.8 - 4.89999999, INTERVAL: 4.9 - 4.99999999, INTERVAL: 5 - 5.09999999, INTERVAL: 5.1 - 5.19999999, INTERVAL: 5.2 - 5.29999999, INTERVAL: 5.3 - 5.39999999, INTERVAL: 5.4 – 5.49999999]  
//Note: does capture 5.4x, by specifying 0.5 offset from base unit

define "Question4x6":  
 expand { Interval[1.55, 5.55 ) } per 0.1  
[INTERVAL: 1.55 - 1.64999999, INTERVAL: 1.65 - 1.74999999, INTERVAL: 1.75 - 1.84999999, INTERVAL: 1.85 - 1.94999999, INTERVAL: 1.95 - 2.04999999, INTERVAL: 2.05 - 2.14999999, INTERVAL: 2.15 - 2.24999999, INTERVAL: 2.25 - 2.34999999, INTERVAL: 2.35 - 2.44999999, INTERVAL: 2.45 - 2.54999999, INTERVAL: 2.55 - 2.64999999, INTERVAL: 2.65 - 2.74999999, INTERVAL: 2.75 - 2.84999999, INTERVAL: 2.85 - 2.94999999, INTERVAL: 2.95 - 3.04999999, INTERVAL: 3.05 - 3.14999999, INTERVAL: 3.15 - 3.24999999, INTERVAL: 3.25 - 3.34999999, INTERVAL: 3.35 - 3.44999999, INTERVAL: 3.45 - 3.54999999, INTERVAL: 3.55 - 3.64999999, INTERVAL: 3.65 - 3.74999999, INTERVAL: 3.75 - 3.84999999, INTERVAL: 3.85 - 3.94999999, INTERVAL: 3.95 - 4.04999999, INTERVAL: 4.05 - 4.14999999, INTERVAL: 4.15 - 4.24999999, INTERVAL: 4.25 - 4.34999999, INTERVAL: 4.35 - 4.44999999, INTERVAL: 4.45 - 4.54999999, INTERVAL: 4.55 - 4.64999999, INTERVAL: 4.65 - 4.74999999, INTERVAL: 4.75 - 4.84999999, INTERVAL: 4.85 - 4.94999999, INTERVAL: 4.95 - 5.04999999, INTERVAL: 5.05 - 5.14999999, INTERVAL: 5.15 - 5.24999999, INTERVAL: 5.25 - 5.34999999, INTERVAL: 5.35 - 5.44999999, INTERVAL: 5.45 - 5.54999999]   
//Note: does capture 5.4x, by specifying 0.55 offset from base unit

//Decimal, Open LO, Closed HI  
define "Question5x":  
 expand { Interval ( 1.0, 5.0]}  
[INTERVAL: 2 - 2, INTERVAL: 3 - 3, INTERVAL: 4 - 4, INTERVAL: 5 - 5]

define "Question5x2":  
 expand { Interval ( 1.0, 5.0]} per 2.0  
[INTERVAL: 2 - 3, INTERVAL: 4 - 5]

define "Question5x3":  
 expand { Interval ( 1.0, 5.0]} per 0.1  
[INTERVAL: 2 - 2.09999999, INTERVAL: 2.1 - 2.19999999, INTERVAL: 2.2 - 2.29999999, INTERVAL: 2.3 - 2.39999999, INTERVAL: 2.4 - 2.49999999, INTERVAL: 2.5 - 2.59999999, INTERVAL: 2.6 - 2.69999999, INTERVAL: 2.7 - 2.79999999, INTERVAL: 2.8 - 2.89999999, INTERVAL: 2.9 - 2.99999999, INTERVAL: 3 - 3.09999999, INTERVAL: 3.1 - 3.19999999, INTERVAL: 3.2 - 3.29999999, INTERVAL: 3.3 - 3.39999999, INTERVAL: 3.4 - 3.49999999, INTERVAL: 3.5 - 3.59999999, INTERVAL: 3.6 - 3.69999999, INTERVAL: 3.7 - 3.79999999, INTERVAL: 3.8 - 3.89999999, INTERVAL: 3.9 - 3.99999999, INTERVAL: 4 - 4.09999999, INTERVAL: 4.1 - 4.19999999, INTERVAL: 4.2 - 4.29999999, INTERVAL: 4.3 - 4.39999999, INTERVAL: 4.4 - 4.49999999, INTERVAL: 4.5 - 4.59999999, INTERVAL: 4.6 - 4.69999999, INTERVAL: 4.7 - 4.79999999, INTERVAL: 4.8 - 4.89999999, INTERVAL: 4.9 - 4.99999999]

//Decimal, Open LO, Open HI  
define "Question6x":  
 expand { Interval ( 1.0, 5.0 ) }  
[INTERVAL: 2 - 2, INTERVAL: 3 - 3, INTERVAL: 4 - 4]

define "Question6x2":  
 expand { Interval ( 1.0, 5.0 ) } per 2.0  
[INTERVAL: 2 - 2.09999999, INTERVAL: 2.1 - 2.19999999, INTERVAL: 2.2 - 2.29999999, INTERVAL: 2.3 - 2.39999999, INTERVAL: 2.4 - 2.49999999, INTERVAL: 2.5 - 2.59999999, INTERVAL: 2.6 - 2.69999999, INTERVAL: 2.7 - 2.79999999, INTERVAL: 2.8 - 2.89999999, INTERVAL: 2.9 - 2.99999999, INTERVAL: 3 - 3.09999999, INTERVAL: 3.1 - 3.19999999, INTERVAL: 3.2 - 3.29999999, INTERVAL: 3.3 - 3.39999999, INTERVAL: 3.4 - 3.49999999, INTERVAL: 3.5 - 3.59999999, INTERVAL: 3.6 - 3.69999999, INTERVAL: 3.7 - 3.79999999, INTERVAL: 3.8 - 3.89999999, INTERVAL: 3.9 - 3.99999999]

define "Question6x4":  
 expand { Interval ( 1.00, 5.00 ) } per 0.1  
[INTERVAL: 2 - 2.09999999, INTERVAL: 2.1 - 2.19999999, INTERVAL: 2.2 - 2.29999999, INTERVAL: 2.3 - 2.39999999, INTERVAL: 2.4 - 2.49999999, INTERVAL: 2.5 - 2.59999999, INTERVAL: 2.6 - 2.69999999, INTERVAL: 2.7 - 2.79999999, INTERVAL: 2.8 - 2.89999999, INTERVAL: 2.9 - 2.99999999, INTERVAL: 3 - 3.09999999, INTERVAL: 3.1 - 3.19999999, INTERVAL: 3.2 - 3.29999999, INTERVAL: 3.3 - 3.39999999, INTERVAL: 3.4 - 3.49999999, INTERVAL: 3.5 - 3.59999999, INTERVAL: 3.6 - 3.69999999, INTERVAL: 3.7 - 3.79999999, INTERVAL: 3.8 - 3.89999999, INTERVAL: 3.9 - 3.99999999]

define "Question6x5":  
 expand { Interval ( 1.50, 5.50 ) } per 0.1  
[INTERVAL: 1.50000001 - 1.6, INTERVAL: 1.60000001 - 1.7, INTERVAL: 1.70000001 - 1.8, INTERVAL: 1.80000001 - 1.9, INTERVAL: 1.90000001 - 2, INTERVAL: 2.00000001 - 2.1, INTERVAL: 2.10000001 - 2.2, INTERVAL: 2.20000001 - 2.3, INTERVAL: 2.30000001 - 2.4, INTERVAL: 2.40000001 - 2.5, INTERVAL: 2.50000001 - 2.6, INTERVAL: 2.60000001 - 2.7, INTERVAL: 2.70000001 - 2.8, INTERVAL: 2.80000001 - 2.9, INTERVAL: 2.90000001 - 3, INTERVAL: 3.00000001 - 3.1, INTERVAL: 3.10000001 - 3.2, INTERVAL: 3.20000001 - 3.3, INTERVAL: 3.30000001 - 3.4, INTERVAL: 3.40000001 - 3.5, INTERVAL: 3.50000001 - 3.6, INTERVAL: 3.60000001 - 3.7, INTERVAL: 3.70000001 - 3.8, INTERVAL: 3.80000001 - 3.9, INTERVAL: 3.90000001 - 4, INTERVAL: 4.00000001 - 4.1, INTERVAL: 4.10000001 - 4.2, INTERVAL: 4.20000001 - 4.3, INTERVAL: 4.30000001 - 4.4, INTERVAL: 4.40000001 - 4.5, INTERVAL: 4.50000001 - 4.6, INTERVAL: 4.60000001 - 4.7, INTERVAL: 4.70000001 - 4.8, INTERVAL: 4.80000001 - 4.9, INTERVAL: 4.90000001 - 5, INTERVAL: 5.00000001 - 5.1, INTERVAL: 5.10000001 - 5.2, INTERVAL: 5.20000001 - 5.3, INTERVAL: 5.30000001 - 5.4]

define "Question6x6":  
 expand { Interval ( 1.55, 5.55 ) } per 0.1  
[INTERVAL: 1.55000001 - 1.65, INTERVAL: 1.65000001 - 1.75, INTERVAL: 1.75000001 - 1.85, INTERVAL: 1.85000001 - 1.95, INTERVAL: 1.95000001 - 2.05, INTERVAL: 2.05000001 - 2.15, INTERVAL: 2.15000001 - 2.25, INTERVAL: 2.25000001 - 2.35, INTERVAL: 2.35000001 - 2.45, INTERVAL: 2.45000001 - 2.55, INTERVAL: 2.55000001 - 2.65, INTERVAL: 2.65000001 - 2.75, INTERVAL: 2.75000001 - 2.85, INTERVAL: 2.85000001 - 2.95, INTERVAL: 2.95000001 - 3.05, INTERVAL: 3.05000001 - 3.15, INTERVAL: 3.15000001 - 3.25, INTERVAL: 3.25000001 - 3.35, INTERVAL: 3.35000001 - 3.45, INTERVAL: 3.45000001 - 3.55, INTERVAL: 3.55000001 - 3.65, INTERVAL: 3.65000001 - 3.75, INTERVAL: 3.75000001 - 3.85, INTERVAL: 3.85000001 - 3.95, INTERVAL: 3.95000001 - 4.05, INTERVAL: 4.05000001 - 4.15, INTERVAL: 4.15000001 - 4.25, INTERVAL: 4.25000001 - 4.35, INTERVAL: 4.35000001 - 4.45, INTERVAL: 4.45000001 - 4.55, INTERVAL: 4.55000001 - 4.65, INTERVAL: 4.65000001 - 4.75, INTERVAL: 4.75000001 - 4.85, INTERVAL: 4.85000001 - 4.95, INTERVAL: 4.95000001 - 5.05, INTERVAL: 5.05000001 - 5.15, INTERVAL: 5.15000001 - 5.25, INTERVAL: 5.25000001 - 5.35, INTERVAL: 5.35000001 - 5.45]

// DateTime Intervals, expand per day, confirms midnight as boundary for day  
define "Question7x":  
 expand { Interval[@2018-01-01, @2018-01-05]} per day  
[INTERVAL: 2018-01-01 - 2018-01-01, INTERVAL: 2018-01-02 - 2018-01-02, INTERVAL: 2018-01-03 - 2018-01-03, INTERVAL: 2018-01-04 - 2018-01-04, INTERVAL: 2018-01-05 - 2018-01-05]

define "Question8x":  
 expand { Interval[@2018-01-01T01:30:00, @2018-01-05T18:30:00]} per day  
[INTERVAL: 01/01/2018 12:00 AM - 01/01/2018 12:00 AM, INTERVAL: 01/02/2018 12:00 AM - 01/02/2018 12:00 AM, INTERVAL: 01/03/2018 12:00 AM - 01/03/2018 12:00 AM, INTERVAL: 01/04/2018 12:00 AM - 01/04/2018 12:00 AM, INTERVAL: 01/05/2018 12:00 AM - 01/05/2018 12:00 AM]  
// Note this example with offset time for discussion in Part B

define "Question9xA":  
 expand { Interval[@2018-01-01, @2018-01-31]} per day  
[INTERVAL: 2018-01-01 - 2018-01-01, INTERVAL: 2018-01-02 - 2018-01-02, INTERVAL: 2018-01-03 - 2018-01-03, INTERVAL: 2018-01-04 - 2018-01-04, INTERVAL: 2018-01-05 - 2018-01-05, INTERVAL: 2018-01-06 - 2018-01-06, INTERVAL: 2018-01-07 - 2018-01-07, INTERVAL: 2018-01-08 - 2018-01-08, INTERVAL: 2018-01-09 - 2018-01-09, INTERVAL: 2018-01-10 - 2018-01-10, INTERVAL: 2018-01-11 - 2018-01-11, INTERVAL: 2018-01-12 - 2018-01-12, INTERVAL: 2018-01-13 - 2018-01-13, INTERVAL: 2018-01-14 - 2018-01-14, INTERVAL: 2018-01-15 - 2018-01-15, INTERVAL: 2018-01-16 - 2018-01-16, INTERVAL: 2018-01-17 - 2018-01-17, INTERVAL: 2018-01-18 - 2018-01-18, INTERVAL: 2018-01-19 - 2018-01-19, INTERVAL: 2018-01-20 - 2018-01-20, INTERVAL: 2018-01-21 - 2018-01-21, INTERVAL: 2018-01-22 - 2018-01-22, INTERVAL: 2018-01-23 - 2018-01-23, INTERVAL: 2018-01-24 - 2018-01-24, INTERVAL: 2018-01-25 - 2018-01-25, INTERVAL: 2018-01-26 - 2018-01-26, INTERVAL: 2018-01-27 - 2018-01-27, INTERVAL: 2018-01-28 - 2018-01-28, INTERVAL: 2018-01-29 - 2018-01-29, INTERVAL: 2018-01-30 - 2018-01-30, INTERVAL: 2018-01-31 – 2018-01-31]  
// DateTime Intervals, expand per day, null HI errors in MAT (memory issue for 9999-12-31)  
  
Well, that was fun. Reread "Question8x" then move on to Part B for further discussion.   
PMuir MD ESAC Standards Team

CQLIT-266 Expand Intervals, Part B re CQLIT-252 CMS249v3 MP2021 20210321

Since CQLIT-266 relates back to CQLIT-252 regarding expand intervals in CMS249v3 for MP2021, we will review the logic from the CQL perspective with CQL snippets from the measure version for MP2021  
library AppropriateUseofDXAScansinWomenUnder65YearsWhoDoNotMeettheRiskFactorProfileforOsteoporoticFracture version '3.2.000'

The 'expand' operator is only found in the "Glucocorticoids Daily Dosage Per Medication" definition which impacts the 'NumberOfRiskFactors' value for the "Has Age Dependent Combination Risk Factors" definition which is a component of "Denominator Exclusions".

**define "Glucocorticoids Daily Dosage Per Medication":**  
 "Glucocorticoid Medication Active" MedicationGlucocorticoid  
 let dosesPerDay: "DosesPerDay"(MedicationGlucocorticoid.frequency),  
 dailyDosage: "GetMedicationDailyDose"(MedicationGlucocorticoid.dosage, dosesPerDay),  
 medPeriod: MedicationGlucocorticoid.relevantPeriod  
 intersect Interval[start of MedicationGlucocorticoid.relevantPeriod,   
 end of "Measurement Period"]  
 return Tuple {  
 code: MedicationGlucocorticoid.code,  
 **daysInPeriod: expand { medPeriod } per day,**  
 dailyDosage: dailyDosage  
 }

The contained definition captures all Glucocorticoids with a start date before the end of Measurement Period.  
**define "Glucocorticoid Medication Active":**  
 ["Medication, Active": "Glucocorticoids (oral only)"] Glucocorticoid  
 where Glucocorticoid.relevantPeriod starts before   
 end of "Measurement Period"

The dosesPerDay variable uses the "DosesPerDay" function to determine the number of doses per day:  
 ... when frequency ~ "Twice a day (qualifier value)" then 2.0 …

The dailyDosage variable uses the "GetMedicationDailyDose" function calculation as the dosage times doses per day for total daily dose:  
define function "GetMedicationDailyDose"(dosage Quantity, dosesPerDay Decimal):  
 dosage \* Quantity { value: dosesPerDay, unit: '/d' }

The medPeriod variable constrains MedicationGlucocorticoid.relevantPeriod to Measurement Period.

**"Glucocorticoids Daily Dosage Per Medication"** returns a tuple containing the medication.code, daysInPeriod and dailyDosage. The 'expand' operator will expand the medPeriod into the individual days from the start of the Medication.relevantPeriod to the end of the relevantPeriod (or Measurement Period, whichever is earlier). As seen from Part A testing, the day boundary is at midnight so that the day is included if there is a medication dosage any time during that specified day.

The following definition uses the 'flatten' operator to combine multiple lists into a single list then returns a list containing the medicationDay and summed totalDailyDosage for that day.

**define "Glucocorticoids Dosage Per Day":** ( distinct ( flatten ( "Glucocorticoids Daily Dosage Per Medication" GlucocorticoidsDailyDosage  
 return all GlucocorticoidsDailyDosage.daysInPeriod  
 )  
 ) ) Day  
 return {  
 medicationDay: Day,  
 totalDailyDosage: Sum("Glucocorticoids Daily Dosage Per Medication" GlucocorticoidsDailyDosagePerDay  
 where Day in GlucocorticoidsDailyDosagePerDay.daysInPeriod  
 return all GlucocorticoidsDailyDosagePerDay.dailyDosage  
 )  
 }

The following definition counts the medicationDays where the summed totalDailyDosage for that day is more than 5 mg/day.

**define "Glucocorticoids Greater Than 5 MG per Day for 90 or More Days":** Count("Glucocorticoids Dosage Per Day" GlucocorticoidsDosage  
 where GlucocorticoidsDosage.totalDailyDosage >= 5 'mg/d'  
 ) >= 90

If more than 90 such days, it returns true which impacts the 'NumberOfRiskFactors' value for the "Has Age Dependent Combination Risk Factors" definition which is a component of "Denominator Exclusions".

The logic could allow a Glucocorticoids medication.relevantPeriod starting before the start of the Medication Period without an end date or during the Medication Period with only a limited supply quantity (eg 5 mg qid prn as directed #20/0 with a relevantPeriod of [@2021-01-05, null] ) to be overestimated as exceeding 90 days; however, this would be mitigated by only providing a 1 instead of a 0 to the 'NumberOfRiskFactors'.

From your initial CQLIT-252 query:   
*After reading the Expand documentation, I am in need of a more details regarding expected behavior.  
For datetime intervals, if the* ***per*** *is a day, for example and a candidate interval starts and ends on non-day boundaries, what would the resulting set of day intervals be?*

From the MAT/Bonnie test environment examples:

define "Question8x":  
 expand { Interval[@2018-01-01T01:30:00, @2018-01-05T18:30:00]} per day  
[INTERVAL: 01/01/2018 12:00 AM - 01/01/2018 12:00 AM, INTERVAL: 01/02/2018 12:00 AM - 01/02/2018 12:00 AM, INTERVAL: 01/03/2018 12:00 AM - 01/03/2018 12:00 AM, INTERVAL: 01/04/2018 12:00 AM - 01/04/2018 12:00 AM, INTERVAL: 01/05/2018 12:00 AM - 01/05/2018 12:00 AM]

Which confirms that midnight is the boundary for the per day expand operation, so that the start/end times of the medication.relevantPeriod would be within the first and last segments but are not the boundary.

I hope this helps clarify the issue.  
PMuir MD ESAC Standards Team

Because this issue involves measure intent, we request the eCQM team review the response and confirm accuracy. >> CQM-4510 EC