# MATGlobalCommonFunctions version 7.0.000

The MATGlobalCommonFunctions library provides common functions and expressions used throughout electronic Clinical Quality Measures (eCQM) published for use in Centers for Medicare and Medicaid (CMS) quality programs.

## Using Quality Data Model (QDM) version '5.6'

## Terminology

valueset "Emergency Department Visit": 'urn:oid:2.16.840.1.113883.3.117.1.7.1.292'

valueset "Encounter Inpatient": 'urn:oid:2.16.840.1.113883.3.666.5.307'

valueset "Intensive Care Unit": 'urn:oid:2.16.840.1.113762.1.4.1029.206'

valueset "Observation Services": 'urn:oid:2.16.840.1.113762.1.4.1111.143'

valueset "Outpatient Surgery Service": 'urn:oid:2.16.840.1.113762.1.4.1110.38'

## Parameters

parameter "Measurement Period" Interval<DateTime>

## Context

Within the Patient context, the results of any retrieve will always be scoped to a single patient, as determined by the environment. Patient-based or Encounter-based results will contain all data for a single patient within the Measurement Period as specified by the query constraints.

context Patient

## Expression ED Encounter

Returns encounters with codes from the "Emergency Department Visit" value set.

define "ED Encounter":

["Encounter, Performed": "Emergency Department Visit"]

## Expression Inpatient Encounter

Returns completed encounters with codes from the "Encounter Inpatient" value set when the inpatient stay is less than or equal to 120 days and the discharge date falls within the Measurement Period.

define "Inpatient Encounter":

["Encounter, Performed": "Encounter Inpatient"] EncounterInpatient

where "LengthInDays"(EncounterInpatient.relevantPeriod)<= 120

and EncounterInpatient.relevantPeriod ends during day of "Measurement Period"

## Function ToDateInterval(Interval<DateTime>) returns Interval<Date>

Returns an interval of date values extracted from the input interval of date-time values.

This function returns an interval constructed using the date from extractor on the start and end values of the input date-time interval. Note that using a precision specifier, such as day of, as part of a timing phrase is preferred to communicate intent to perform day-level comparison, as well as for general readability. e.g. Input Attribute 2022-02-01T00:00:00.000+00:00, Output Return 2022-02-01 (ISO-8601 format YYYY-MM-DDTHH:MM:SS.msZ where Z is timezone offset -05:00 EST New York)

Unless otherwise specified, DateTime and Time comparisons (including interval operations on intervals of DateTime or Time) in CQL are performed to millisecond precision. Be aware that datetime comparison involving different timezone offsets with a precision of less than 1 day granularity (e.g. hours) will normalize the timezone offset to that of the evaluation request timezone stamp. This may impact edge cases for boundaries such as Measurement Period, daylight savings transitions or across different time zone regions.

define function "ToDateInterval"(period Interval<DateTime>):

Interval[date from start of period, date from end of period]

### Examples:

For example, the function could be used to get day-level comparison for an interval from a year before the start of the Measurement Period to the end of the Measurement Period as in the following. However, note that the function must still be used on both sides of the during operator.

define Example1:

["Physical Exam, Performed": "Observation Services"] RetinalExam

where "ToDateInterval"("NormalizeInterval"(RetinalExam.relevantDatetime, RetinalExam.relevantPeriod))

during "ToDateInterval"(Interval[start of "Measurement Period" - 1 year, end of "Measurement Period"])

## Function LengthInDays(Interval<DateTime>) returns Integer

Returns the number of days between the start and the end of a given interval. Calculates the difference in calendar days between the start and end of the given interval. Difference calculations are performed by truncating the datetime values at the next precision and then performing the corresponding duration calculation on the truncated values (midnights crossed) e.g. [2022-02-01T23:00,2022-02-02T01:00] becomes [2022-02-01,2022-02-02] then returns 1 day.

define function "LengthInDays"(Value Interval<DateTime> ):

difference in days between start of Value and end of Value

## Function HospitalizationLocations("Encounter, Performed") returns List<Location>

Returns a list of all facility locations within an encounter, including locations immediately prior to the Emergency Department visit.

define function "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 }

## Function EmergencyDepartmentArrivalTime("Encounter, Performed") returns DateTime

Returns the documented date and time of patient arrival for an Emergency Department visit.

define function "EmergencyDepartmentArrivalTime"(Encounter "Encounter, Performed" ):

start of First(("HospitalizationLocations"(Encounter))HospitalLocation

where HospitalLocation.code in "Emergency Department Visit"

sort by start of locationPeriod

).locationPeriod

## Function HospitalAdmissionTime("Encounter, Performed") returns DateTime

Returns patient admission date and time for an inpatient encounter or for an Emergency Department visit completed immediately prior to the inpatient admission.

define function "HospitalAdmissionTime"(Encounter "Encounter, Performed" ):

start of "Hospitalization"(Encounter)

## Function HospitalArrivalTime("Encounter, Performed") returns DateTime

Returns documented patient arrival date and time for an inpatient encounter or for an Emergency Department visit completed immediately prior to the inpatient admission.

define function "HospitalArrivalTime"(Encounter "Encounter, Performed" ):

start of First(("HospitalizationLocations"(Encounter))HospitalLocation

sort by start of locationPeriod

).locationPeriod

## Function Hospitalization("Encounter, Performed") returns Interval<DateTime>

Returns the total date and time interval from the start of the inpatient encounter or Emergency Department visit completed immediately prior to the inpatient encounter through to the end of the inpatient stay. The Coalesce operator returns the first non-null expression from the associated attributes.

define function "Hospitalization"(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 Interval[Coalesce(start of EDVisit.relevantPeriod, start of Visit.relevantPeriod),

end of Visit.relevantPeriod]

## Function HospitalizationLengthOfStay("Encounter, Performed") returns Integer

Returns the number of days from the start of the inpatient encounter or Emergency Department visit completed immediately prior to the inpatient encounter through to the end of the inpatient stay. The LengthInDays function returns the difference in calendar days, including leap years, as midnights crossed.

define function "HospitalizationLengthofStay"(Encounter "Encounter, Performed" ):

LengthInDays("Hospitalization"(Encounter))

## Function HospitalDepartureTime("Encounter, Performed") returns DateTime

Returns the last date and time of patient departure from facility location associated with the completed inpatient encounter including an immediately prior Emergency Department visit.

define function "HospitalDepartureTime"(Encounter "Encounter, Performed" ):

end of Last(("HospitalizationLocations"(Encounter))HospitalLocation

sort by start of locationPeriod

).locationPeriod

## Function HospitalDischargeTime("Encounter, Performed") returns DateTime

Returns the discharge date and time for a completed inpatient encounter.

define function "HospitalDischargeTime"(Encounter "Encounter, Performed" ):

end of Encounter.relevantPeriod

## Function HospitalizationWithObservationAndOutpatientSurgeryService("Encounter, Performed") returns Interval<DateTime>

Returns the total date and time interval from the start to end of a completed inpatient encounter including dates and times of observation or outpatient services that occurred immediately prior to the inpatient encounter. The Coalesce operator returns the first non-null expression from the associated attributes.

define function "HospitalizationWithObservationAndOutpatientSurgeryService"(Encounter "Encounter, Performed" ):

Encounter Visit

let ObsVisit: Last(["Encounter, Performed": "Observation Services"] LastObs

where LastObs.relevantPeriod ends 1 hour or less on or before start of Visit.relevantPeriod

sort by

end of relevantPeriod

),

VisitStart: Coalesce(start of ObsVisit.relevantPeriod, start of Visit.relevantPeriod),

EDVisit: Last(["Encounter, Performed": "Emergency Department Visit"] LastED

where LastED.relevantPeriod ends 1 hour or less on or before VisitStart

sort by

end of relevantPeriod

),

VisitStartWithED: Coalesce(start of EDVisit.relevantPeriod, VisitStart),

OutpatientSurgeryVisit: Last(["Encounter, Performed": "Outpatient Surgery Service"] LastSurgeryOP

where LastSurgeryOP.relevantPeriod ends 1 hour or less on or before VisitStartWithED

sort by

end of relevantPeriod

)

return Interval[Coalesce(start of OutpatientSurgeryVisit.relevantPeriod, VisitStartWithED),

end of Visit.relevantPeriod]

## Function HospitalizationWithObservation("Encounter, Performed") returns Interval<DateTime>

Returns the total date and time interval from start to end of a completed inpatient encounter including dates and times of Observation and Emergency Department visits that occurred immediately prior to the inpatient encounter. The Coalesce operator returns the first non-null expression from the associated attributes.

define function "HospitalizationWithObservation"(Encounter "Encounter, Performed" ):

Encounter Visit

let ObsVisit: Last(["Encounter, Performed": "Observation Services"] LastObs

where LastObs.relevantPeriod ends 1 hour or less on or before start of Visit.relevantPeriod

sort by

end of relevantPeriod

),

VisitStart: Coalesce(start of ObsVisit.relevantPeriod, start of Visit.relevantPeriod),

EDVisit: Last(["Encounter, Performed": "Emergency Department Visit"] LastED

where LastED.relevantPeriod ends 1 hour or less on or before VisitStart

sort by

end of relevantPeriod

)

return Interval[Coalesce(start of EDVisit.relevantPeriod, VisitStart),

end of Visit.relevantPeriod]

## Function HospitalizationWithObservationLengthofStay("Encounter, Performed") returns Interval<DateTime>

Returns the number of days of a given hospitalization from the start of any immediately prior Emergency Department visit through the associated Observation visit to the discharge of the completed inpatient encounter. The LengthInDays function returns the difference in calendar days, including leap years, as midnights crossed.

define function "HospitalizationWithObservationLengthofStay"(Encounter "Encounter, Performed" ):

"LengthInDays"("HospitalizationWithObservation"(Encounter))

## Function FirstInpatientIntensiveCareUnit("Encounter, Performed") returns Location

Returns the first intensive care unit for the given encounter, without considering any immediately prior Emergency Department visit.

define function "FirstInpatientIntensiveCareUnit"(Encounter "Encounter, Performed" ):

First((Encounter.facilityLocations)HospitalLocation

where HospitalLocation.code in "Intensive Care Unit"

and HospitalLocation.locationPeriod during Encounter.relevantPeriod

sort by start of locationPeriod

)

## Function NormalizeInterval(DateTime, Interval<DateTime>) returns Interval<DateTime>

Given a datetime and a period, returns the period (if a period is provided) or the interval beginning and ending on the datetime (if a datetime is provided). This allows evaluation of EHR data elements which might be stored as either DateTime or Period.

define function "NormalizeInterval"(pointInTime DateTime, period Interval<DateTime> ):

if pointInTime is not null then Interval[pointInTime, pointInTime]

else if period is not null then period

else null as Interval<DateTime>

## Function HasStart(Interval<DateTime>) returns Boolean

Given an interval, returns true if the interval has a starting boundary specified (i.e. the start of the interval is not null and not the minimum DateTime value). Function evaluates for an empty start of interval by checking for the presence of the lowest possible value. Inclusive [closed] boundaries, indicated with square brackets, could include null as beginning of interval. Exclusive (open) boundaries, indicated with parentheses, could exclude null as lowest granularity for the minimum DateTime value for the interval (i.e. null+1 unit of granularity).

define function "HasStart"(period Interval<DateTime> ):

not ( start of period is null

or start of period = minimum DateTime

)

## Function HasEnd(Interval<DateTime>) returns Boolean

Given an interval, returns true if the interval has an ending boundary specified (i.e. the end of the interval is not null and not the maximum DateTime value). Function evaluates for an empty end of interval by checking for the presence of the highest possible value. Inclusive [closed] boundaries, indicated with square brackets, could include null (infinity) as end of interval. Exclusive (open) boundaries, indicated with parentheses, could exclude null (infinity) as highest granularity for the maximum DateTime value for the interval (i.e. infinity-1 unit of granularity).

define function "HasEnd"(period Interval<DateTime> ):

not (

end of period is null

or

end of period = maximum DateTime

)

## Function Latest(Interval<DateTime>) returns DateTime

Returns the latest date and time from a given interval as the ending point if the interval has an ending boundary specified, otherwise returns the starting point.

define function "Latest"(period Interval<DateTime> ):

if ( HasEnd(period)) then

end of period

else start of period

## Function Earliest(Interval<DateTime>) returns DateTime

Returns the earliest date and time from a given interval as the starting point if the interval has a starting boundary specified, otherwise returns the ending point.

define function "Earliest"(period Interval<DateTime> ):

if ( HasStart(period)) then start of period

else

end of period

## Function LatestOf(DateTime, Interval<DateTime>) returns DateTime

Returns the last chronological date and time for a data element whether stored as a datetime or period. Depending upon the data available, priority is datetime, ending point of interval, then starting point of interval.

define function "LatestOf"(pointInTime DateTime, period Interval<DateTime> ):

Latest(NormalizeInterval(pointInTime, period))

## Function EarliestOf(DateTime, Interval<DateTime>) returns DateTime

Returns the first chronological date and time for a data element whether stored as a datetime or period. Depending upon the data available, priority is datetime, starting point of interval, then ending point of interval.

define function "EarliestOf"(pointInTime DateTime, period Interval<DateTime> ):

Earliest(NormalizeInterval(pointInTime, period))