CMS
CENTERS FOR MEDICARE & MEDICAID SERVICES Centers for Medicare & Medicaid Services

Clinical Quality Language (CQL) Style Guide

Version 8.0

May 2024

The Centers for Medicare & Medicaid Services (CMS) contracted with Mathematica and its partners, including The Joint Commission, to develop this style guide to support the electronic specification and maintenance of electronic clinical quality measures that eligible clinicians, eligible hospitals, and critical access hospitals can use for reporting using certified electronic health records (EHRs) for CMS quality programs. Mathematica and its partners developed this document under two Measure and Instrument Development and Support indefinite delivery, indefinite quantity master contract vehicle task order contracts: Electronic Clinical Quality Measures Development and Maintenance for Eligible Clinicians (CMS Contract #75FCMC18D0032, Task Order #75FCMC19F0004) and Behavioral Health Measures Development and Inpatient and Outpatient Measure Maintenance (CMS Contract #75FCMC18D0032, Task Order #75FCMC19F0003).

CONTENTS

1. BACKGROUND

Clinical Quality Language (CQL)[1] is an HL7 standard developed as part of the Clinical Quality Framework (CQF) initiative. CQL is intended to be a clinically focused, author-friendly, and human-readable language that promotes consistency and harmonization across the CQF standards.[2]

Users of this style guide are assumed to have a baseline knowledge of CQL. Please refer to the following links for more information on CQL:

- Electronic Clinical Quality Improvement (eCQI) Resource Center
- CQL Formatting and Usage Wiki

This guide was created by under the Electronic Clinical Quality Measures Development and Maintenance for Eligible Clinicians contract and the Behavioral Health Measures Development and Inpatient and Outpatient Measure Maintenance contract, in coordination with the Centers for Medicare & Medicaid Services (CMS). It is based on guidance from the CQL Formatting and Usage Wiki. The purpose of the guide is to promote consistency and reusability of the specifications in measures included in federal reporting programs. Measure developers and interested parties contributed to the development of the standards in this guide.

The Measure Authoring Tool (MAT) supports the authoring of CQL and sharing of CQL libraries. Please see the MAT User Guide for additional MAT-specific information.

[1] The guidance in this document is based on CQL STU 1.5.2, located at https://cql.hl7.org/01-introduction.html.

[2] Raw CQL files are human readable, but there is also an HTML version in the electronic clinical quality measure package exported from the MAT. The HTML human readable provides a view consistent with the style of earlier electronic clinical quality measures based on Quality Data Model logic.

2. INTRODUCTION

The twofold purpose of this style guide is to (1) standardize the expression of measure concepts across electronic clinical quality measures (eCQMs) and (2) define a uniform “look and feel” for eCQM logic using CQL. The guide focuses on common best practices implemented across CQL-based eCQMs in CMS reporting programs. It also promotes the use of consistent language within the framework of CQL, including libraries, aliases, definitions, and functions, and it provides guidance on other conventions, such as operator precedence. Measure stewards or developers who are developing or specifying eCQMs for potential inclusion in CMS reporting programs should follow these best practices.

This guide applies to Health Quality Measures Format measures; a separate guide will be defined for Fast Healthcare Interoperable Resources (FHIR)-based measures when appropriate.

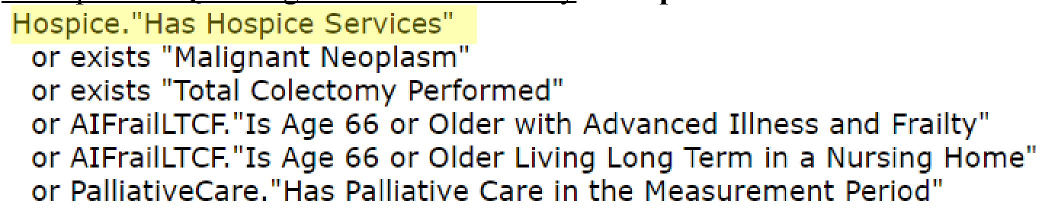
3. STANDARDS: LIBRARIES

Libraries, the basic units of sharing CQL, consist of a foundation of CQL statements used within an eCQM. Every eCQM has at least one CQL library called the "primary" library. eCQMs can use CQL expressions stored in shared libraries. Shared libraries enable similar logic to be used across multiple measures, improving standardization and harmonization across eCQMs. Shared libraries can exist at the **local** or **global** level in the MAT.

- ***Local* shared library:** All MAT users can view CQL libraries, but they are typically shared and editable only among a specific group of users within the MAT. A local library should be used when similar functions or definitions are used across measures; it is required if five or more measures are affected.

  The local CQL library **'Hospice'** is used in the following CQL definition statement. The **'Hospice'** shared library contains a CQL definition named "Has Hospice Services." This example shows how shared libraries, when used, are represented in measure logic.

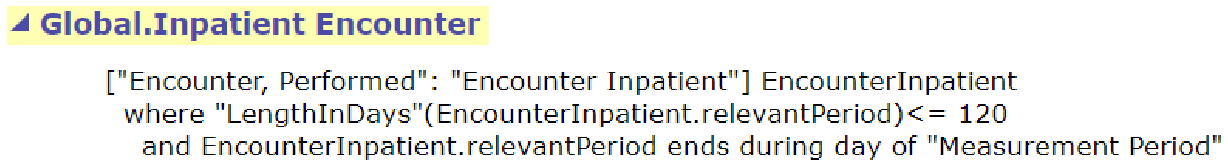
  Example of CQL using a local shared library:[3] **Hospice**



```
Hospice."Has Hospice Services"
  or exists "Malignant Neoplasm"
  or exists "Total Colectomy Performed"
  or AIFrailLTCF."Is Age 66 or Older with Advanced Illness and Frailty"
  or AIFrailLTCF."Is Age 66 or Older Living Long Term in a Nursing Home"
  or PalliativeCare."Has Palliative Care in the Measurement Period"
```

- ***Global* shared library:** A global library is a shared CQL library created in the MAT that contains CQL expressions for all measure developers to use when specifying an eCQM. All MAT users can access the global common library, MATGlobalCommonFunctionsQDM.[4] The global common library is updated and published as needed to align with the Quality Data Model (QDM) and CQL standards incorporated into the MAT. Using this global common library reduces duplication and maintains consistency of measure logic across measure specifications. Users may submit questions about the MAT Global Common Library to the Office of the National Coordinator for Health Information Technology (ONC) CQL Issue Tracker.

  Example of CQL in the global common library:[5]



```
◢ Global.Inpatient Encounter

      ["Encounter, Performed": "Encounter Inpatient"] EncounterInpatient
        where "LengthInDays"(EncounterInpatient.relevantPeriod)<= 120
          and EncounterInpatient.relevantPeriod ends during day of "Measurement Period"
```

[3] Example from CMS130v12 for 2024 reporting.

[4] See MAT Global Common Functions Library (GCL) v8.0 at https://ecqi.healthit.gov/sites/default/files/MATGlobalCommonFunctions-2024-05.zip.

[5] Example from CMS506v6 for 2024 reporting.

A. Best practices for naming CQL libraries

The measure name and measure library name do not have to be the same. When naming a new measure CQL library, adhere to the following standards:

- **DO** use PascalCase, or the capitalization of the first letter of each compound word in a variable.[6]

  Example: **`AdultOutpatientEncounters`**

- **DO** use names that are short, descriptive, and easy to read and that accurately reflect the contents of the library.

  Example: **`Hospice`**

- **DO NOT** start the library name with a special character,[7] number, or underscore. The following is an example to avoid:

  Example*:* `*AdultEncounters`

- **DO NOT** use spaces or special characters[8] in the library name. The following is an example to avoid:

  Example: `Adult+PediatricEncounter`

[6] See Appendix B for a complete list of case-type definitions.

[7] Special characters are symbols such as +, _, -, *, and /. This list is non-exhaustive.

[8] Special characters are symbols such as +, _, -, *, and /. This list is non-exhaustive.

4. STANDARDS: DEFINITIONS

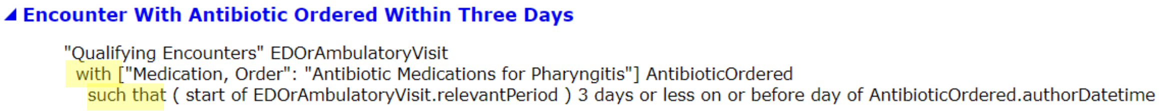
Definitions are concise logical CQL expressions that describe the meaning of measure concepts. Measure population logic also references definitions. Measure developers should reuse and reference definitions in other CQL expressions, whenever appropriate.

A. Best practices for writing CQL definitions

When writing definitions, use the following best practices:

- **DO** use a 'with' or 'without' statement with a 'such that' statement when comparing two different data types or data sources.

  Example of CQL definition using 'with':[9]

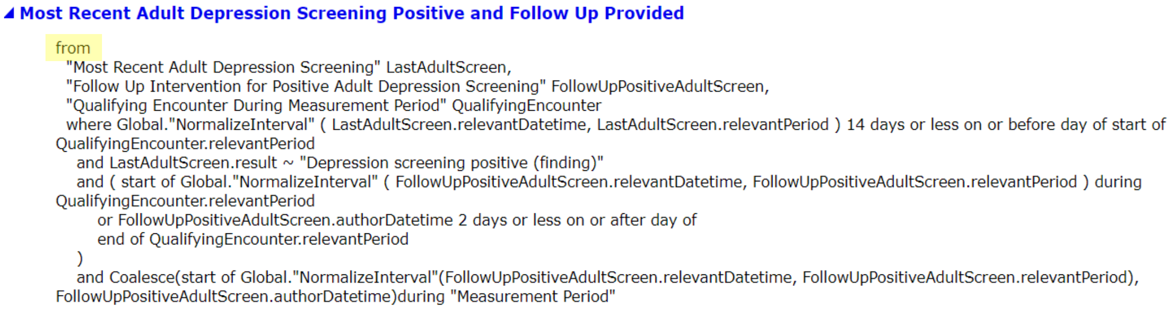


◢ Encounter With Antibiotic Ordered Within Three Days

```
"Qualifying Encounters" EDOrAmbulatoryVisit
  with ["Medication, Order": "Antibiotic Medications for Pharyngitis"] AntibioticOrdered
    such that ( start of EDOrAmbulatoryVisit.relevantPeriod ) 3 days or less on or before day of AntibioticOrdered.authorDatetime
```

- **DO** use a 'from' statement when comparing more than two sources of information.

  Example of CQL definition using 'from':[10]



◢ Most Recent Adult Depression Screening Positive and Follow Up Provided

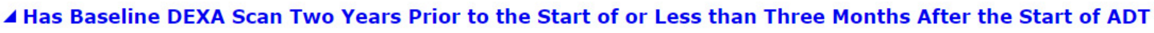
```
from
  "Most Recent Adult Depression Screening" LastAdultScreen,
  "Follow Up Intervention for Positive Adult Depression Screening" FollowUpPositiveAdultScreen,
  "Qualifying Encounter During Measurement Period" QualifyingEncounter
  where Global."NormalizeInterval" ( LastAdultScreen.relevantDatetime, LastAdultScreen.relevantPeriod ) 14 days or less on or before day of start of
QualifyingEncounter.relevantPeriod
    and LastAdultScreen.result ~ "Depression screening positive (finding)"
    and ( start of Global."NormalizeInterval" ( FollowUpPositiveAdultScreen.relevantDatetime, FollowUpPositiveAdultScreen.relevantPeriod ) during
QualifyingEncounter.relevantPeriod
        or FollowUpPositiveAdultScreen.authorDatetime 2 days or less on or after day of
        end of QualifyingEncounter.relevantPeriod
    )
    and Coalesce(start of Global."NormalizeInterval"(FollowUpPositiveAdultScreen.relevantDatetime, FollowUpPositiveAdultScreen.relevantPeriod),
FollowUpPositiveAdultScreen.authorDatetime)during "Measurement Period"
```

B. Best practices for naming CQL definitions

When naming definitions, use the following best practices:

- **DO** use Title Case, or capitalization of the first and last word and all major words in between.[11]

  Example of CQL definition name:[12]



◢ Has Baseline DEXA Scan Two Years Prior to the Start of or Less than Three Months After the Start of ADT

[9] Example from CMS146v12 for 2024 reporting.

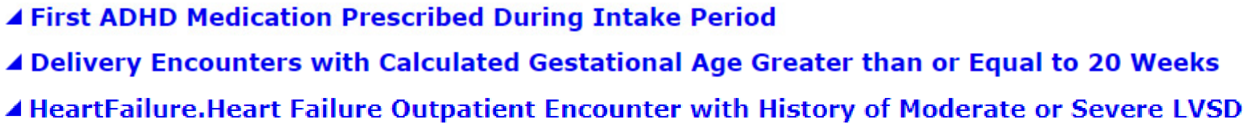
[10] Example from CMS2v13 for 2024 reporting.

[11] See Appendix B for a complete list of case-type definitions.

[12] Example from CMS645v7 for 2024 reporting.

- **DO** create names that are easy to read, describe the contents of the logical expression, and provide context as to what makes the statement unique and clinically meaningful.

  Examples of CQL definition names:[13]



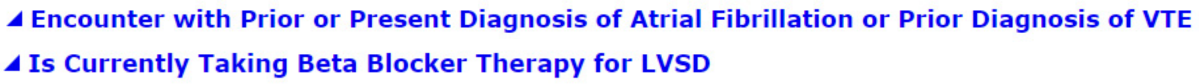
◢ First ADHD Medication Prescribed During Intake Period

◢ Delivery Encounters with Calculated Gestational Age Greater than or Equal to 20 Weeks

◢ HeartFailure.Heart Failure Outpatient Encounter with History of Moderate or Severe LVSD

- **DO** use only abbreviations or acronyms that are consistent with terminology used in the measure's narrative.

  Examples of CQL definition names:[14]

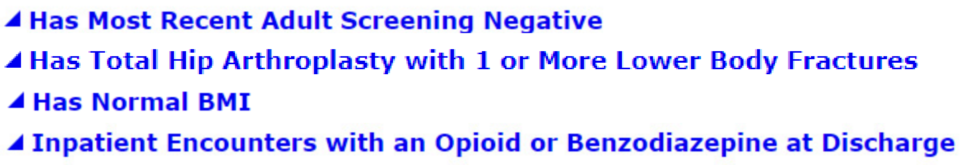


◢ Encounter with Prior or Present Diagnosis of Atrial Fibrillation or Prior Diagnosis of VTE

◢ Is Currently Taking Beta Blocker Therapy for LVSD

- **DO** create definition names that are clear and indicate the return.[15] For example, a yes/no return should be named like a question, using the words "Is" or "Has."

  Examples of CQL definition names:[16]



◢ Has Most Recent Adult Screening Negative

◢ Has Total Hip Arthroplasty with 1 or More Lower Body Fractures

◢ Has Normal BMI

◢ Inpatient Encounters with an Opioid or Benzodiazepine at Discharge

- **DO NOT** give a definition the same name as a value set,[17] as this can reduce the clarity and readability of the logic and raise the risk of logical errors. The following is an example to avoid.

  Example:

```
Cognitive Assessment
 ["Intervention, Performed": "Cognitive Assessment"]
```

- **DO NOT** give a definition the same name as a CQL operator.[18] For example, 'Union' is a CQL operator used to combine all elements from multiple lists of values. Do not use 'Union' as a name for a definition statement.

[13] Examples from CMS136v13, CMS1028v2, and CMS135v12 for 2024 reporting.

[14] Examples from CMS108v12 and CMS144v12 for 2024 reporting.

[15] The MAT will display the "Return Type" in the definition form (when there are no CQL errors).

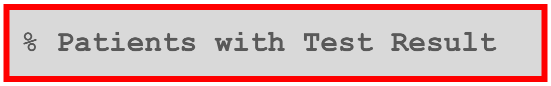
[16] Examples from CMS2v13, CMS56v12, CMS69v12, and CMS506v6 for 2024 reporting.

[17] In CQL, a definition name and value set name cannot share the same identifier.

[18] See the CQL Specification for a full list of operators: https://cql.hl7.org/STU4/02-authorsguide.html#operations.

- **DO NOT** use special characters[19] in definition names. The following is an example to avoid.

  Example:



```
% Patients with Test Result
```

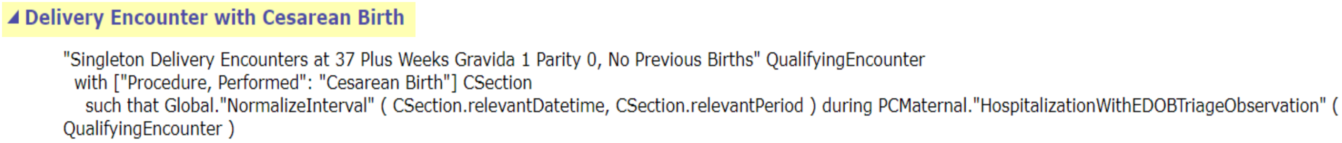
Use Table 1 as a guide for naming definitions. The left column provides examples of definition names, and the right column shows alternatives that are clearer and more readable.

Table 1. Making good definition names better and clearer

Good definition name	Better definition name
Anticoagulant Not Given at Discharge	Reason for Not Giving Anticoagulant at Discharge
In Demographic	Single Live Birth Encounter with Gestational Age 37 Weeks or More
Lab Test with Result	Most Recent Elevated HbA1c with Result

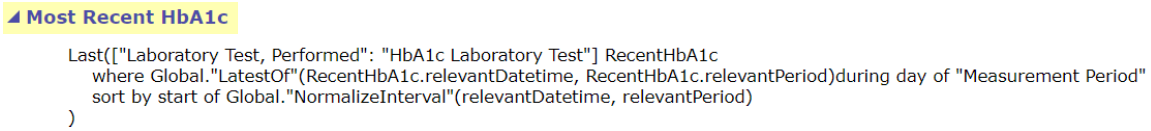
The following are more examples of CQL definitions that are descriptive and clinically meaningful.

Example of CQL definition:[20]



```
◢ Delivery Encounter with Cesarean Birth

    "Singleton Delivery Encounters at 37 Plus Weeks Gravida 1 Parity 0, No Previous Births" QualifyingEncounter
      with ["Procedure, Performed": "Cesarean Birth"] CSection
        such that Global."NormalizeInterval" ( CSection.relevantDatetime, CSection.relevantPeriod ) during PCMaternal."HospitalizationWithEDOBTriageObservation" (
    QualifyingEncounter )
```

Example of CQL definition:[21]



```
◢ Most Recent HbA1c

    Last(["Laboratory Test, Performed": "HbA1c Laboratory Test"] RecentHbA1c
        where Global."LatestOf"(RecentHbA1c.relevantDatetime, RecentHbA1c.relevantPeriod)during day of "Measurement Period"
        sort by start of Global."NormalizeInterval"(relevantDatetime, relevantPeriod)
    )
```

[19] Special characters are symbols such as +, _, -, *, and /. This list is non-exhaustive.

[20] Example from CMS334v5 for 2024 reporting.

[21] Example from CMS122v12 for 2024 reporting.

C. Standards for naming definitions across measures

Use Table 2 as a guide for naming definitions that use common concepts across measures. The left column presents the concept, and the right column shows the recommended standard naming convention.

Table 2. Standard definition names for use across measures

Concept	Recommended definition name
Hospice Exclusions Exclusions for Hospice	Has Hospice Services
Encounters *or* Eligible Encounters *or* Valid Encounters	Qualifying Encounters (Note: generally used for clinician measures)

5. STANDARDS: ALIASES OR ARGUMENT NAMES

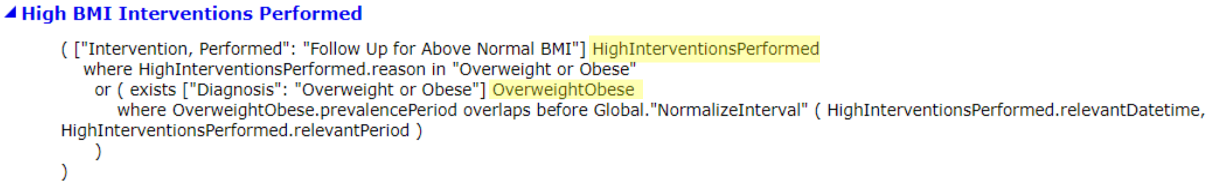
Aliases, or argument names, are identifiers that refer to individual CQL expressions or libraries. Aliases should correlate clearly to their source and can be reused to avoid restating key expressions. Alias names should maintain their meaning and uniformity within and across measures. This allows for a more fluid, concise, and standardized CQL expression. Measure developers can create aliases for libraries, functions, and definitions.

A. Best practices for using CQL aliases and argument names

When naming aliases and argument names, use the following standards:

- **DO** use PascalCase.[22]

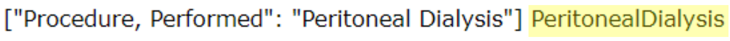
  Examples of CQL aliases:[23]



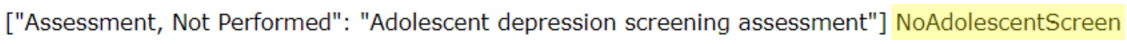
```
◢ High BMI Interventions Performed
    ( ["Intervention, Performed": "Follow Up for Above Normal BMI"] HighInterventionsPerformed
      where HighInterventionsPerformed.reason in "Overweight or Obese"
        or ( exists ["Diagnosis": "Overweight or Obese"] OverweightObese
            where OverweightObese.prevalencePeriod overlaps before Global."NormalizeInterval" ( HighInterventionsPerformed.relevantDatetime,
    HighInterventionsPerformed.relevantPeriod )
        )
    )
```

- **DO** use names that are short, descriptive, and easy to read and that accurately reflect the identified concept.

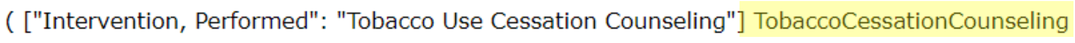
  Examples of CQL aliases:[24]

1) 

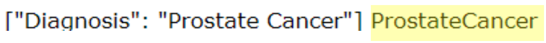
```
["Procedure, Performed": "Peritoneal Dialysis"] PeritonealDialysis
```

2) 

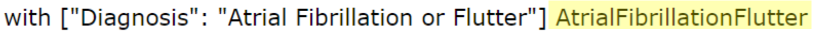
```
["Assessment, Not Performed": "Adolescent depression screening assessment"] NoAdolescentScreen
```

3) 

```
( ["Intervention, Performed": "Tobacco Use Cessation Counseling"] TobaccoCessationCounseling
```

4) 

```
["Diagnosis": "Prostate Cancer"] ProstateCancer
```

5) 

```
with ["Diagnosis": "Atrial Fibrillation or Flutter"] AtrialFibrillationFlutter
```

- **DO** use names that clearly distinguish two different events that have the same logic but occur at different times.

  For example, consider logic that is looking for a lab result that occurred before and after a clinical action.

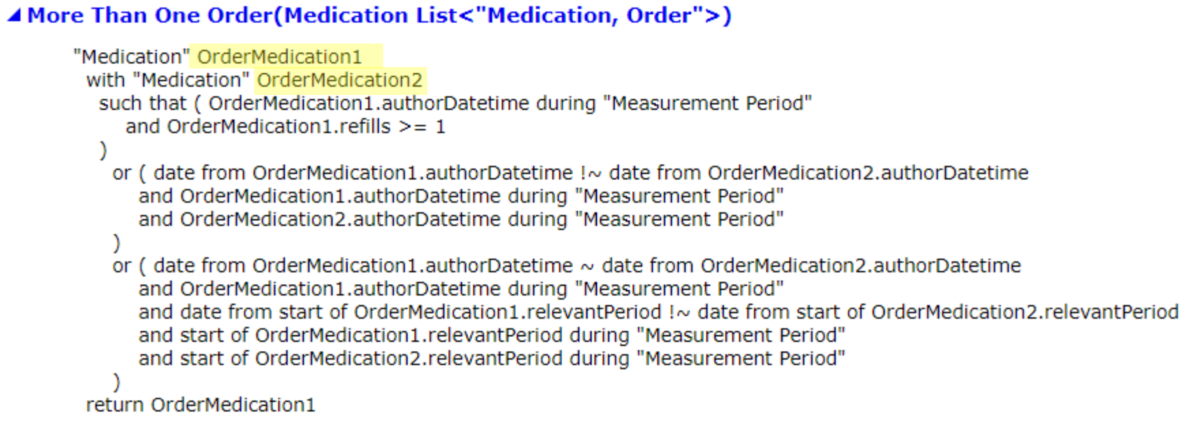
Good alias name	Better alias name
Lab1	LabWithResultBeforeProcedure
Lab2	LabWithResultAfterProcedure

[22] See Appendix B for a complete list of case-type definitions.

[23] Examples from CMS69v12 for 2024 reporting.

[24] Examples from CMS147v13, CMS2v13, CMS138v12, CMS129v13, and CMS71v13 for 2024 reporting.

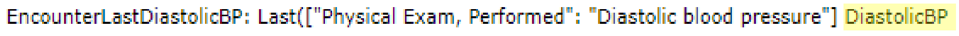
Examples of CQL aliases:[25]



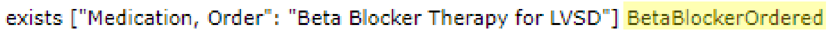
```
◢ More Than One Order(Medication List<"Medication, Order">)

        "Medication" OrderMedication1
          with "Medication" OrderMedication2
            such that ( OrderMedication1.authorDatetime during "Measurement Period"
                and OrderMedication1.refills >= 1
            )
              or ( date from OrderMedication1.authorDatetime !~ date from OrderMedication2.authorDatetime
                  and OrderMedication1.authorDatetime during "Measurement Period"
                  and OrderMedication2.authorDatetime during "Measurement Period"
              )
              or ( date from OrderMedication1.authorDatetime ~ date from OrderMedication2.authorDatetime
                  and OrderMedication1.authorDatetime during "Measurement Period"
                  and date from start of OrderMedication1.relevantPeriod !~ date from start of OrderMedication2.relevantPeriod
                  and start of OrderMedication1.relevantPeriod during "Measurement Period"
                  and start of OrderMedication2.relevantPeriod during "Measurement Period"
              )
          return OrderMedication1
```

- **DO** create alias names that are clinically focused.

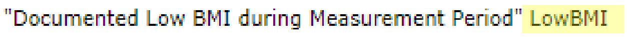
  Examples of CQL aliases:[26]



```
EncounterLastDiastolicBP: Last(["Physical Exam, Performed": "Diastolic blood pressure"] DiastolicBP
```



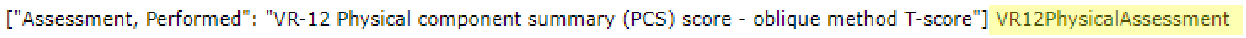
```
exists ["Medication, Order": "Beta Blocker Therapy for LVSD"] BetaBlockerOrdered
```



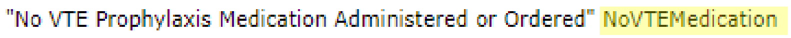
```
"Documented Low BMI during Measurement Period" LowBMI
```

- **DO** use only abbreviations or acronyms that are consistent with terminology used in the measure's narrative sections.

  Examples of CQL aliases (Note: in each example, the acronym is defined in the narrative.):[27]



```
["Assessment, Performed": "VR-12 Physical component summary (PCS) score - oblique method T-score"] VR12PhysicalAssessment
```



```
"No VTE Prophylaxis Medication Administered or Ordered" NoVTEMedication
```

- **DO NOT** reuse the same alias for different data elements, as this can reduce the clarity and readability of the logic.

  See the following for an example to avoid. The alias **`HeartRate`** is reused in two different definitions in the same measure, each with a different scope.

[25] Example from CMS156v12 for 2024 reporting.

[26] Examples from CMS22v12, CMS144v12, and CMS69v12 for 2024 reporting.

[27] Examples from CMS56v12 and CMS108v12 for 2024 reporting.

Example:

```
First(["Encounter, Performed": "Heart Rate Visit"] HeartRate
       with ["Diagnosis": "Essential Hypertension"]
       Hypertension such that HeartRate.relevantPeriod overlaps
       Hypertension.prevalencePeriod)
```

```
Last(["Physical Exam, Performed": "Heart Rate Exam"] HeartRate
       with "Initial Blood Pressure Visit" InitialEncounter
       such that HeartRate.relevantDatetime during
       InitialEncounter.relevantPeriod
       where HeartRate.result is not null
       sort by start of relevantDatetime

    )
```

- **DO NOT** give an alias the same name as the definition or value set title, as this can reduce the clarity and readability of the logic and raise the risk of logical errors.
- **DO NOT** use an alias if the definition statement does not require additional logic, as it is unnecessary and does not add any value.

Use Table 3 as a guide for naming aliases. The left column lists examples of alias names that measure developers should avoid. The alternatives in the right column are clearer and more readable.

Table 3. Making aliases easier to read and more clinically focused

Alias names to avoid	Better alias names
D *or* Dx *or* Diagnosis	HeartFailure Pregnancy Asthma Bradycardia
Med Medication	BetaBlockerOrdered AntidepressantAdministered
P *or* Proc *or* Procedure	CardiacSurgery Dialysis
Lab *or* LabTest	HepBAntigenTest MumpsTiter PregnancyTest
E *or* Enc	Encounter (use with caution if referring to several types of encounters in a measure) InpatientEncounter HeartFailureEncounter Psychotherapy
["Physical Exam, Performed": "Heart Rate"] **Exam**	HeartRateExam
["Diagnostic Study, Performed": "Ejection Fraction"] **Study**	EjectionFractionStudy

Use Table 4 as a guide for further improving alias names. The left column lists examples of aliases. The right column shows alternatives that are clearer and more descriptive.

Table 4. Making good alias names more descriptive

Good alias name	Better alias name
HeartRate	FirstHeartRate
AntithromboticNotGiven	NoAntithrombotic
VisualExam	VisualFootExam
Fracture	LowerBodyFracture

Use Table 5 as a guide for creating distinctions between two aliases that have similar characteristics within a measure by adding specificity.

Table 5. Differentiating between aliases with similar concepts by adding specificity

Similar concepts	Similar alias names with specificity
Heart failure encounter *and* Heart failure diagnosis	HeartFailureEncounter *and* HeartFailureDiagnosis

B. Standards for naming aliases across measures

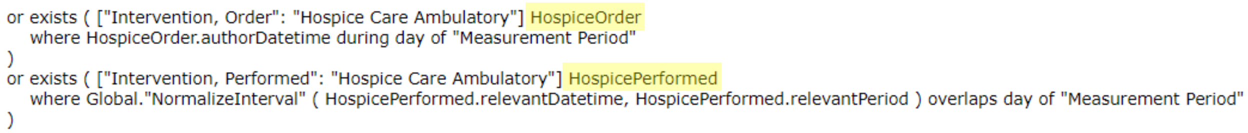
Use Table 6 as a guide for naming aliases that use common concepts across measures. The left column presents the alias concept, and the right column presents the recommended standard alias naming convention.

Table 6. Standard alias names recommended for use across measures

Concept	Standardized alias
Hospice discharge	DischargeToHospice
Hospice care order	HospiceOrder
Hospice intervention performed	HospicePerformed

See the following examples of an alias name.

Examples of CQL aliases:[28] **`HospiceOrder and HospicePerformed`**

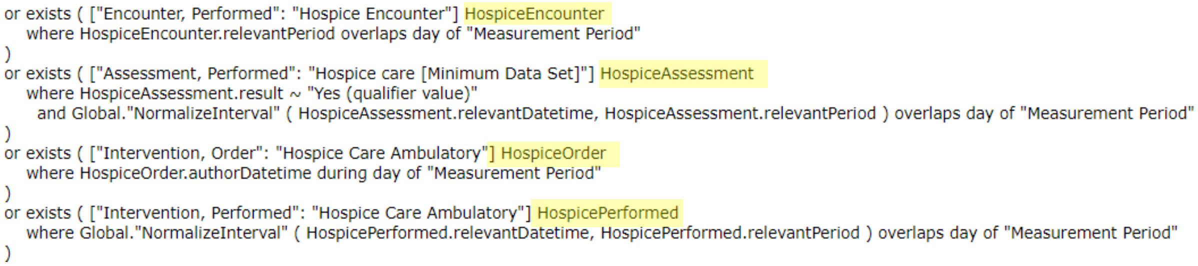


```
or exists ( ["Intervention, Order": "Hospice Care Ambulatory"] HospiceOrder
   where HospiceOrder.authorDatetime during day of "Measurement Period"
)
or exists ( ["Intervention, Performed": "Hospice Care Ambulatory"] HospicePerformed
   where Global."NormalizeInterval" ( HospicePerformed.relevantDatetime, HospicePerformed.relevantPeriod ) overlaps day of "Measurement Period"
)
```

Accurate CQL alias names can aid in reading logic that models similar concepts. See the following example that uses the concept of hospice.

[28] Example from CMS125v12 for 2024 reporting.

Examples of CQL aliases:[29] **`HospiceEncounter, HospiceAssessment, HospiceOrder, HospicePerformed`**



```
or exists ( ["Encounter, Performed": "Hospice Encounter"] HospiceEncounter
   where HospiceEncounter.relevantPeriod overlaps day of "Measurement Period"
)
or exists ( ["Assessment, Performed": "Hospice care [Minimum Data Set]"] HospiceAssessment
   where HospiceAssessment.result ~ "Yes (qualifier value)"
     and Global."NormalizeInterval" ( HospiceAssessment.relevantDatetime, HospiceAssessment.relevantPeriod ) overlaps day of "Measurement Period"
)
or exists ( ["Intervention, Order": "Hospice Care Ambulatory"] HospiceOrder
   where HospiceOrder.authorDatetime during day of "Measurement Period"
)
or exists ( ["Intervention, Performed": "Hospice Care Ambulatory"] HospicePerformed
   where Global."NormalizeInterval" ( HospicePerformed.relevantDatetime, HospicePerformed.relevantPeriod ) overlaps day of "Measurement Period"
)
```

[29] Example from CMS125v12 for 2024 reporting.

6. FUNCTIONS

A function is a named CQL expression that can perform any variety of calculations. Before creating new functions, measure developers should review and—to the extent possible and applicable—use the predefined functions available in the MAT or in the shared global common library. Functions act on the input arguments passed to them, whereas definitions operate only on the expressions in the definition.

A. Selecting functions

To differentiate similar functions, choose from the predefined list of functions in the MAT or from the global common library. Select the function that is most appropriate to meet the measure's intent. See the following examples of preferred functions from the global common library.

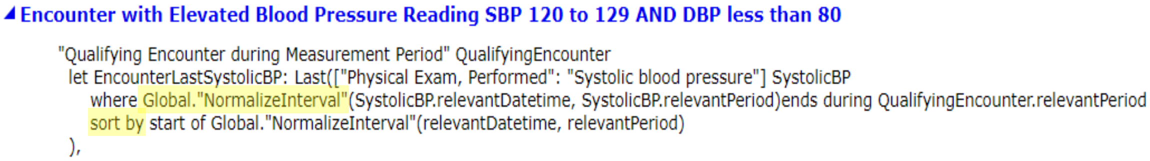
- "NormalizeInterval" function:[30]

  Example of CQL function: **`Global CQL function`**

```
Global."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>
```

  - Use the "NormalizeInterval" function for QDM datatypes that have use cases for both a relevantDatetime and a relevantPeriod, which will reduce the implementation burden associated with variable use of timing attributes across measures.[31]
  - The "NormalizeInterval" function might be needed when using a sort clause.

  Example of CQL function:[32] **`NormalizeInterval with sort`**



Encounter with Elevated Blood Pressure Reading SBP 120 to 129 AND DBP less than 80

```
"Qualifying Encounter during Measurement Period" QualifyingEncounter
 let EncounterLastSystolicBP: Last(["Physical Exam, Performed": "Systolic blood pressure"] SystolicBP
    where Global."NormalizeInterval"(SystolicBP.relevantDatetime, SystolicBP.relevantPeriod)ends during QualifyingEncounter.relevantPeriod
    sort by start of Global."NormalizeInterval"(relevantDatetime, relevantPeriod)
 ),
```

[30] For more information on the "NormalizeInterval" function, please review the "Teach Me CQL" series at https://ecqi.healthit.gov/new-resources-available-teach-me-clinical-quality-language-cql-video-series.

[31] Review the QDM documentation on the eCQI Resource Center for a full list of QDM datatypes and their attributes.

[32] Example from CMS22v12 for 2024 reporting.

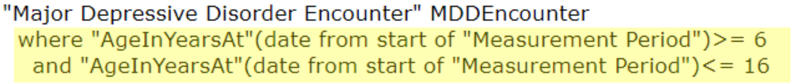
- Age functions:

  Example of CQL function: **`AgeInYearsAt`**

  `"AgeInYearsAt"``(date from start of)`

  - The **`AgeInYearsAt`** function calculates age using **birth date** and time. By adding the 'date from' within the parenthetical statement, the measure developer can augment the function to not use time in the calculation.

  Example of CQL function (used in logic):[33]



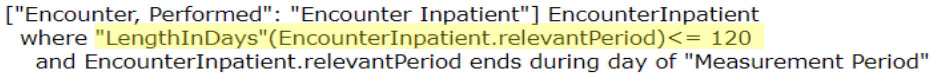
```
"Major Depressive Disorder Encounter" MDDEncounter
 where "AgeInYearsAt"(date from start of "Measurement Period")>= 6
  and "AgeInYearsAt"(date from start of "Measurement Period")<= 16
```

- Length-of-stay functions (generally used for hospital measures):

  Example of CQL function: **`Global."LengthInDays"()`**

  - **`LengthInDays()`** calculates the difference in calendar days between the start and end of the given interval. Timing intervals should always be noted in chronological order as [start, finish] to avoid negative time intervals.
  - This function can be used to calculate the length of a hospital stay for an inpatient encounter, from admission to discharge. Also see the following examples.

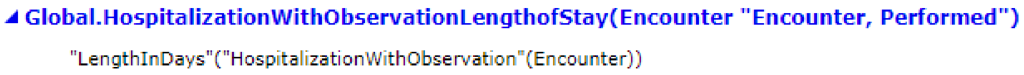
  Example of CQL function (used in logic):[34]



```
["Encounter, Performed": "Encounter Inpatient"] EncounterInpatient
  where "LengthInDays"(EncounterInpatient.relevantPeriod)<= 120
    and EncounterInpatient.relevantPeriod ends during day of "Measurement Period"
```

  Example of CQL function:[35]
  `Global."HospitalizationWithObservationLengthOfStay"()`

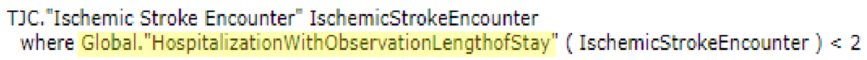
  - Returns the length in days from the start of any immediately prior emergency department visit through the observation visit to the discharge for the given encounter.



```
◢ Global.HospitalizationWithObservationLengthofStay(Encounter "Encounter, Performed")
      "LengthInDays"("HospitalizationWithObservation"(Encounter))
```

  Example of CQL function (used in logic):[36]

  `Global."HospitalizationWithObservationLengthofStay"()`



```
TJC."Ischemic Stroke Encounter" IschemicStrokeEncounter
  where Global."HospitalizationWithObservationLengthofStay" ( IschemicStrokeEncounter ) < 2
```

[33] Example from CMS177v12 for 2024 reporting.

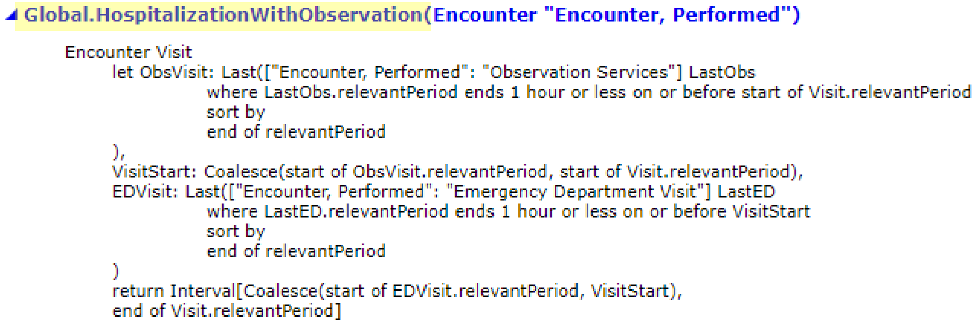
[34] Example from CMS108v12 for 2024 reporting.

[35] Example from CMS72v12 for 2024 reporting.

[36] Example from CMS72v12 for 2024 reporting.

Example of CQL function:[37] **`Global."HospitalizationWithObservation"()`**

- This function returns the total interval for (1) admission to observation for the given encounter or (2) admission of any immediately prior emergency department visit to the observation for the given encounter.



```
Global.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]
```

B. Best practices and standards for naming new CQL functions

New function names should be short, descriptive, and easy to read and should provide an expression that accurately represents the identified concept. When naming functions, use the following standards:

- **DO** use PascalCase.[38]

  Example of CQL function name:[39]

  `LastHistoryPretermBirth(Encounter "Encounter, Performed")`

- **DO** use spaces after commas to separate arguments.

  Example of CQL function:

  `Global.NormalizeInterval(pointInTime DateTime, period Interval<DateTime>)`

- **DO NOT** give a function the same name as a CQL predefined function, as this can raise the risk of logical errors.

  Process step: Please review the predefined operators available in the MAT[40] to ensure the name of your function is not the same.

[37] Example from CMS71v13 for 2024 reporting.

[38] See Appendix B for complete definitions of case types.

[39] Example from CMS334v5 for 2024 reporting.

[40] See the MAT user guide for a list of operators: https://www.emeasuretool.cms.gov/sites/default/files/2022-11/MAT%20User%20Guide%20v6.10%20FHIR%2011042022.pdf.

7. OTHER CQL BEST PRACTICES

A. Population criteria

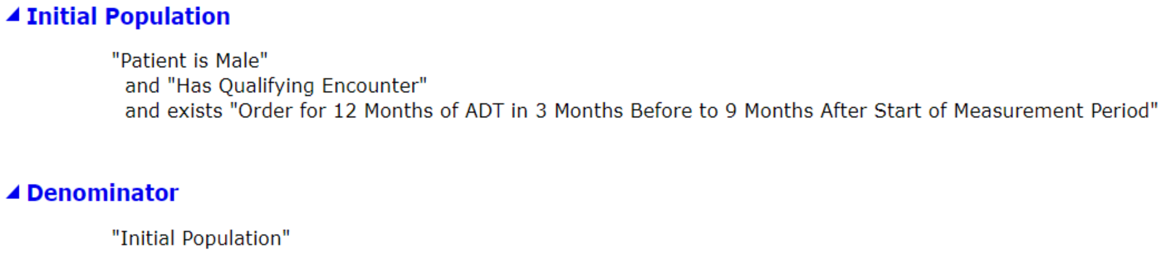
When using population criteria, be descriptive and specific, making sure names are easy to read. The following example shows how to improve the naming of population criteria.

Current population criteria	Improved population criteria

`Initial Population`
`'In Demographic'`
	`Initial Population`
	`'Single Live Birth Encounter with Gestational Age 37 Weeks or More'`

When the denominator population criteria are equivalent to the initial population criteria, state "Initial Population" for the Denominator, as shown below:

Example CQL:[41] Initial Population = Denominator



Initial Population

```
"Patient is Male"
  and "Has Qualifying Encounter"
  and exists "Order for 12 Months of ADT in 3 Months Before to 9 Months After Start of Measurement Period"
```

Denominator

```
"Initial Population"
```

- **DO** use brackets, **[** and **]**, to represent a closed interval and parentheses, **(** and **)**, for open intervals. This pertains to definitions containing age intervals to avoid inconsistencies and to harmonize logic across measures. The CQL Specification states that "intervals in CQL are represented by specifying the low and high points of the Interval and whether the boundary is inclusive (meaning the boundary point is part of the interval) or exclusive (meaning the boundary point is excluded from the interval). Following standard mathematics notation, inclusive (closed) boundaries are indicated with square brackets, and exclusive (open) boundaries are indicated with parentheses."[42]
- Consider the following statement:

  ```
  Interval[3, 5)
  ```

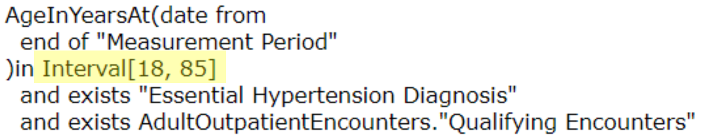
  This expression results in an `Interval` that contains the integers 3 and 4 because the [represents a closed or inclusive boundary. But the `Interval` does not contain 5 because the) represents an open or exclusive boundary.

[41] Example from CMS645v7 for 2024 reporting.

[42] See the CQL Specification for more on interval values: https://cql.hl7.org/02-authorsguide.html#interval-values.

In the following example, the definition uses `Interval[18, 85]` to include patients ages 18 to 85 but exclude patients ages 17 and younger and 86 and older at the end of the measurement period.

Example CQL:[43] **Age Interval**



```
AgeInYearsAt(date from
  end of "Measurement Period"
)in Interval[18, 85]
  and exists "Essential Hypertension Diagnosis"
  and exists AdultOutpatientEncounters."Qualifying Encounters"
```

For measure development, avoid using mixed-boundary notation in the same logical statement.

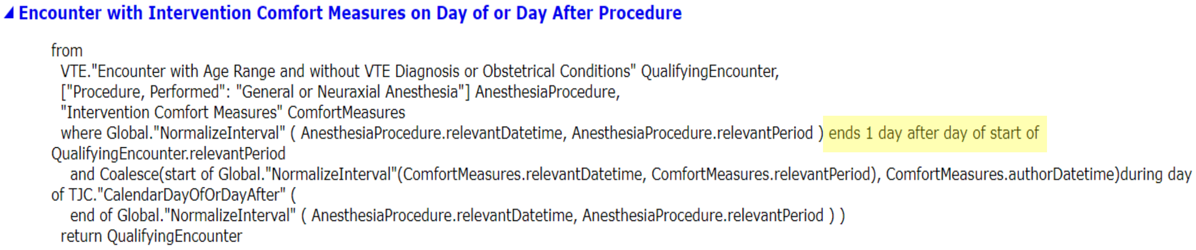
B. Additional timing phrases

CQL supports precision-based date/time comparisons. Be sure to consider whether date or time is being evaluated in timing phrases. Additional timing phrases might be needed when making a statement such as "A starts/ends before/after or concurrent with start of B." These additional timing phrases indicate a time precision in the logic and ignore the day component.

When comparing date-time valued elements to the measurement period, unless time-sensitive comparison is truly desired, use the **day of** modifier to indicate that the comparison should be performed to the day level of precision.

Example CQL:[44] ends 1 day after day of start of

A supplementary timing constraint is added to ensure the timing of the relevant period for the AnesthesiaProcedure ends 1 day after the start of relevant period of the QualifyingEncounter.

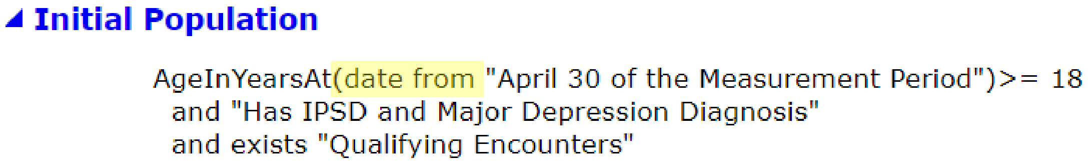


```
◢ Encounter with Intervention Comfort Measures on Day of or Day After Procedure

    from
      VTE."Encounter with Age Range and without VTE Diagnosis or Obstetrical Conditions" QualifyingEncounter,
      ["Procedure, Performed": "General or Neuraxial Anesthesia"] AnesthesiaProcedure,
      "Intervention Comfort Measures" ComfortMeasures
      where Global."NormalizeInterval" ( AnesthesiaProcedure.relevantDatetime, AnesthesiaProcedure.relevantPeriod ) ends 1 day after day of start of
    QualifyingEncounter.relevantPeriod
        and Coalesce(start of Global."NormalizeInterval"(ComfortMeasures.relevantDatetime, ComfortMeasures.relevantPeriod), ComfortMeasures.authorDatetime)during day
    of TJC."CalendarDayOfOrDayAfter" (
        end of Global."NormalizeInterval" ( AnesthesiaProcedure.relevantDatetime, AnesthesiaProcedure.relevantPeriod ) )
      return QualifyingEncounter
```

[43] Example from CMS165v12 for 2024 reporting.

[44] Example from CMS108v12 for 2024 reporting.

In addition, when assessing date-time valued elements for comparison to the measurement period, unless time-sensitive comparison is truly desired, use the **date from** operator to access only the date portion of the date-time valued element. In the following example, the **date from** operator is used to compare the birth date at the date level of precision.[45]



◢ Initial Population

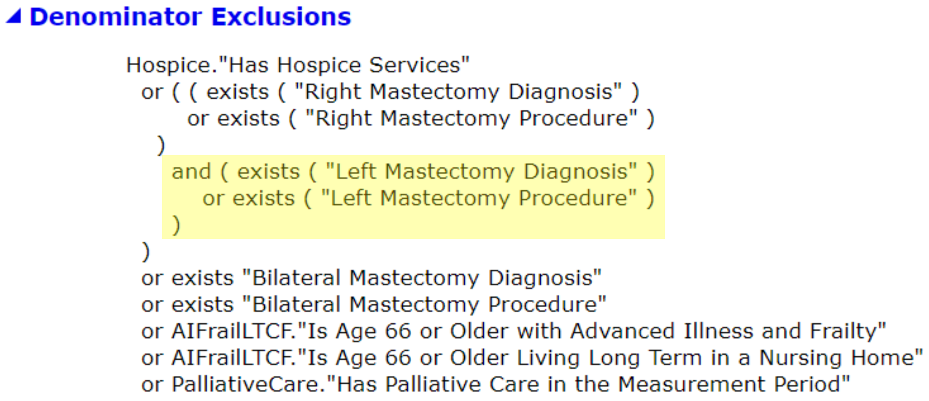
```
AgeInYearsAt(date from "April 30 of the Measurement Period")>= 18
  and "Has IPSD and Major Depression Diagnosis"
  and exists "Qualifying Encounters"
```

C. Operator precedence

Precedence in CQL expressions is determined by the order of appearance in the expression, left to right. To ensure consistent and predictable behavior in the order of operations in CQL expressions, use parentheses around a grouping to enforce higher precedence. See the table in Appendix A for more details.

In the example below, parentheses are used to promote operator precedence around **exists "Left Mastectomy Diagnosis"** and **exists "Left Mastectomy Procedure"** and to make the groupings clear.

Example CQL:[46] Operator precedence



◢ Denominator Exclusions

```
Hospice."Has Hospice Services"
  or ( ( exists ( "Right Mastectomy Diagnosis" )
       or exists ( "Right Mastectomy Procedure" )
    )
      and ( exists ( "Left Mastectomy Diagnosis" )
          or exists ( "Left Mastectomy Procedure" )
      )
  )
  or exists "Bilateral Mastectomy Diagnosis"
  or exists "Bilateral Mastectomy Procedure"
  or AIFrailLTCF."Is Age 66 or Older with Advanced Illness and Frailty"
  or AIFrailLTCF."Is Age 66 or Older Living Long Term in a Nursing Home"
  or PalliativeCare."Has Palliative Care in the Measurement Period"
```

D. Direct reference codes

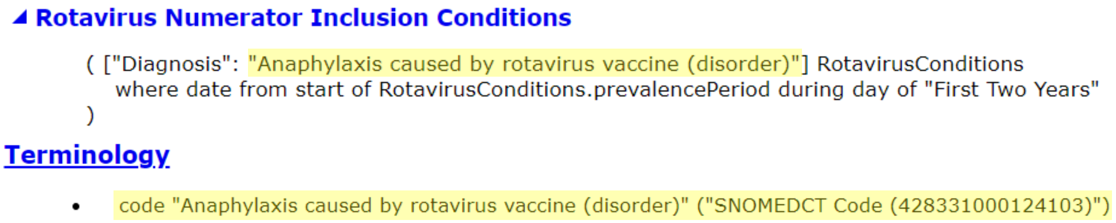
Measure developers maintain and publish value sets on the Value Set Authority Center web site. Value sets and codes are listed in the terminology section of each measure specification. Direct reference codes (DRCs) are single codes that can be referenced directly in CQL logic, instead of creating a value set containing a single code. DRCs are recommended for all single

[45] Example from CMS128v12 for 2024 reporting.

[46] Example from CMS125v12 for 2024 reporting.

codes and may be used for other single-use terminology codes.[47] When included in a definition, a DRC is incorporated in the CQL syntax through use of the code descriptor. The DRC's specific code and corresponding descriptor will always be included in the Terminology section of the human readable; it might also be referenced in the Data Criteria section if used as part of a QDM element, not just as an attribute of a previously defined QDM element.

Example:[48] DRC used in CQL logic



◢ Rotavirus Numerator Inclusion Conditions

```
( ["Diagnosis": "Anaphylaxis caused by rotavirus vaccine (disorder)"] RotavirusConditions
    where date from start of RotavirusConditions.prevalencePeriod during day of "First Two Years"
)
```

Terminology

- code "Anaphylaxis caused by rotavirus vaccine (disorder)" ("SNOMEDCT Code (428331000124103)")

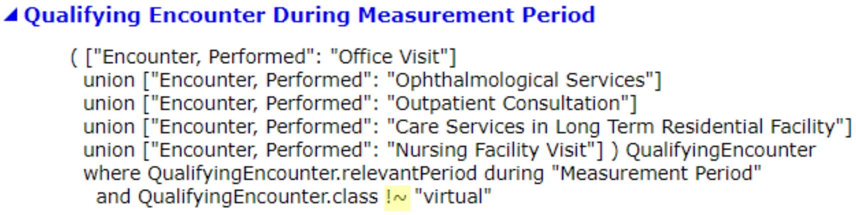
E. Other style considerations

Using the following best practices will enhance the readability and usability of measure specifications:

- Do NOT copy and paste bullets into the MAT header text. The bullets will not format correctly. Instead, use hyphens ("-") to improve the readability of the text. If including citations for clinical recommendations, consider replacing bullet formatting with hyphens after moving text into the MAT header to improve readability. **Any Word-based formatting can cause issues when pasted into MAT header text.**

- Use the equivalence operator (~) to indicate equivalence between two concepts. Use the inequivalence operator (!~) to indicate inequivalence between two concepts.

Example:[49]



◢ Qualifying Encounter During Measurement Period

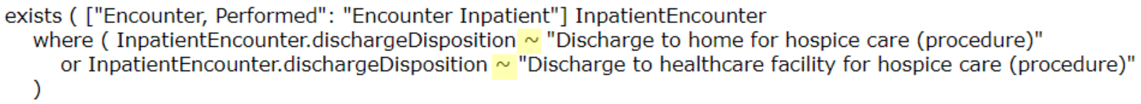
```
( ["Encounter, Performed": "Office Visit"]
  union ["Encounter, Performed": "Ophthalmological Services"]
  union ["Encounter, Performed": "Outpatient Consultation"]
  union ["Encounter, Performed": "Care Services in Long Term Residential Facility"]
  union ["Encounter, Performed": "Nursing Facility Visit"] ) QualifyingEncounter
  where QualifyingEncounter.relevantPeriod during "Measurement Period"
    and QualifyingEncounter.class !~ "virtual"
```

[47] Codes, Code Systems, and Value Sets (2022): https://mmshub.cms.gov/sites/default/files/Codes-Code-Systems-Value-Sets.pdf.

[48] Example from CMS117v12 for 2024 reporting.

[49] Example from CMS142v12 for 2024 reporting.

Example:[50]



```
exists ( ["Encounter, Performed": "Encounter Inpatient"] InpatientEncounter
  where ( InpatientEncounter.dischargeDisposition ~ "Discharge to home for hospice care (procedure)"
    or InpatientEncounter.dischargeDisposition ~ "Discharge to healthcare facility for hospice care (procedure)"
  )
```

[50] Example from CMS146v12 for 2024 reporting.

VERSION HISTORY

Version	Date	Description of change
2.0	August 21, 2018	Initial publication
3.0	May 2019	Removed references related to retired QDM logic
		Updated examples of logic to reflect most recent use
		Added examples of logic in each section to provide context
		Added clarifying language to content in each section
		Updated language and content to align with standards changes QDM 5.4 and CQL 1.3
		Removed reference to Keyword-Distinct
		Updated examples using birthdate to reflect the addition of birth date, a direct reference code
4.0	May 2020	Updated language and content to align with standards changes QDM 5.5 and CQL 1.4
		Updated examples of logic to reflect most recent use
5.0	May 2021	Removed references related to retired QDM logic
		Updated examples of logic to reflect most recent use
		Added section on best practices for writing definitions
		Added guidance on creating age intervals
		Added ‘NormalizeInterval’ function to the ‘Selecting functions’ section
6.0	May 2022	Updated examples of logic to reflect most recent use
7.0	May 2023	Updated examples of logic to reflect most recent use
8.0	May 2024	Updated examples of logic to reflect most recent use

APPENDIX A:

THE ORDER OF OPERATOR PRECEDENCE IN CQL

Table A.1. Order of operator precedence in CQL (highest to lowest)

Category	Operators
Primary	. [] ()
Conversion Phrase	**“convert..to”**
Unary Arithmetic	unary +/-
Extractor	**“start end difference duration width successor predecessor of”** *component* **“singleton from”**
Exponentiation	^
Multiplicative	* / **“div mod”**
Additive	+ - &
Conditional	**“if..then..else** **case..else..end”**
Unary List	**“distinct collapse flatten expand”**
Unary Test	**“is null true false”**
Type Operators	**“is as cast..as”**
Unary Logical	**“not exists”**
Between	**“between”** *precision* **“between”** **“duration in”** *precision* **“between”** **“difference in”** *precision* **“between”**
Comparison	<= < > >=
Timing Phrase	**“same as** **includes** **during** **before/after** **within”**
Interval Operators	**“meets overlaps starts ends”**
Equality	= != ~ !~
Membership	**“in contains”**
Conjunction	**“and”**
Disjunction	**“or xor”**
Implication	**“implies”**
Binary List	**“union intersect except”**

Source: https://cql.hl7.org/03-developersguide.html#operator-precedence.

Note: Bolded text enclosed by quotes is used to identify CQL operator language.

APPENDIX B:

CASE-TYPE DEFINITIONS

Case-Type Definitions

(Note: CQL is a case-sensitive language.)

- **lowercase** – All letters are lowercase.
- **camelCase** – First letters of words are capitalized except for the first word, with no whitespace characters allowed (used for QDM attributes).
- **PascalCase** – First letters of words are capitalized, including words not capitalized in Title Case such as “and” and “of,” with no whitespace characters allowed.
- **Title Case** – Standard title casing including spaces and tabs, but no other whitespace characters allowed.