



Centers for Medicare & Medicaid Services

Quality Data Model (QDM)-based Clinical Quality Language (CQL) Style Guide

Version 10.0

May 2026

The Centers for Medicare & Medicaid Services (CMS) contracted with Mathematica and its partners, including Joint Commission, to develop this style guide to support the electronic specification and maintenance of electronic clinical quality measures (eCQMs) that eligible clinicians and hospitals can use for reporting using certified electronic health records (EHRs) for CMS quality programs. Mathematica and its partners developed this document under a Measure and Instrument Development and Support indefinite delivery, indefinite quantity master contract vehicle task order: Behavioral Health Measures Development & Inpatient and Outpatient Measures Maintenance (CMS Contract 75FCMC18D0032, Task Order 75FCMC24F0136).

CONTENTS

1. Background.....	1
2. Introduction.....	2
3. Standards: Libraries	3
A. Best practices for naming primary CQL libraries	4
B. Best practices for naming shared CQL libraries.....	4
4. Standards: Definitions	5
A. Best practices for writing CQL definitions	5
B. Best practices for naming CQL definitions.....	6
C. Standards for naming definitions across measures.....	8
5. Standards: Aliases.....	9
A. Best practices for using CQL aliases and argument names.....	9
6. Functions.....	12
A. Selecting functions	12
B. Best practices and standards for naming CQL functions	15
7. Other CQL Best Practices.....	16
A. Population criteria	16
B. Additional timing phrases.....	17
C. Operator precedence.....	18
D. Value sets.....	19
E. Direct reference codes	20
F. Other style considerations	20
Version History.....	22
Appendix A: The Order of Operator Precedence In CQL.....	A.1
Appendix B: Case Type Definitions.....	B.1

TABLES

Table 1. Making good definition names better and clearer7

Table 2. Standard definition names for use across measures.....8

Table 3. Making aliases easier to read and more clinically focused.....11

Table 4. Making good alias names more descriptive.....11

Table 5. Differentiating between aliases with similar concepts by adding specificity.....11

Table A.1. Order of operator precedence in CQL (highest to lowest).....A.2

1. BACKGROUND

Clinical Quality Language (CQL)¹ is a Health Level Seven (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 resources for more information on CQL:

- [The CQL page of the Electronic Clinical Quality Improvement \(eCQI\) Resource Center](#)
- [CQL Formatting and Usage Wiki](#) and [CQL Basics Cheat Sheet](#)
- Monthly Cooking with CQL, Quality Data Model (QDM), and Fast Healthcare Interoperability Resources (FHIR) webinars
 - Additional information on the webinar series is located on the [eCQI Resource Center's Calendar page](#).

This guide is maintained under the Behavioral Health Measures Development & Inpatient and Outpatient Measures Maintenance contract in coordination with the Centers for Medicare & Medicaid Services (CMS). It is based on guidance from the [CQL Formatting and Usage Wiki](#) and [CQL Basics Cheat Sheet](#). 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 Development Integrated Environment (MADiE) tool supports the authoring of CQL and sharing of CQL libraries. Please see the [MADiE User Guide located on the Training & Resources page](#) for additional MADiE-specific information.

¹ The guidance in this document is based on CQL version 1.5.3, located at <https://cql.hl7.org/01-introduction.html>.

² Raw CQL files are human readable. There is also a HyperText Markup Language (HTML) version in the electronic clinical quality measure package exported from MADiE.

2. INTRODUCTION

This guide applies to electronic clinical quality measures (eCQMs). A separate guide will be defined for FHIR-based measures.

The twofold purpose of this style guide is to (1) standardize the expression of measure concepts across eCQMs using CQL and (2) define a uniform look and feel for eCQM logic using CQL. The guide focuses on common CQL best practices implemented across eCQMs used in CMS reporting programs. This guide 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.

Please note that the CQL examples provided throughout this guide draw from the eCQM specifications posted for the 2026 reporting/performance period. Unless otherwise indicated, the examples of CQL logic in this guide show how the logic is displayed in a measure's HTML human-readable file.

3. STANDARDS: LIBRARIES

Libraries are collections of CQL statements that are the basic units for sharing CQL logic. Every eCQM has at least one CQL library, which is called the primary library and contains the logic for a specific eCQM. eCQMs can also use CQL expressions stored in shared libraries. Shared libraries enable logic to be used across multiple measures, improving standardization and harmonization across eCQMs. Shared libraries can exist at the **local** or **global** level.

- **Global shared library.** A global shared library contains CQL expressions that facilitate consistent handling of data elements within the data model. A global library is a shared CQL library that contains CQL expressions for all measure developers to use when specifying an eCQM. All MADiE users can access the global shared library, **CQMCommonQDM**, for which developers commonly use an alias of “Global”.³ This library is updated and published as needed to align with the QDM and CQL standards incorporated into MADiE. Using this global shared library reduces duplication and maintains consistency of measure logic across measure specifications. Users may submit questions about the **CQMCommonQDM** library to the [Office of the National Coordinator for Health Information Technology \(ONC\) CQL Issue Tracker](#).
- **Local shared library.** A local shared library is created by a measure developer for use within their own measures. Though some local shared libraries are applicable only to that developer’s own measures, other local shared libraries may be more widely applicable and could also be used in measures maintained by other developers. A local shared library should be used when similar functions or definitions are repeated across measures; it is required if five or more measures are affected.

The measure developer can include a shared library in their measure and will usually create an alias used to reference the library in a measure’s CQL logic expressions; for example, the alias used for the local shared library, **HospiceQDM**, is “Hospice”.

The example below shows how a library’s alias is defined in a measure’s CQL primary library file:⁴

```
include HospiceQDM version '7.0.000' called Hospice
```

The **HospiceQDM** local shared library contains a CQL definition named, “Has Hospice Services.” The example below shows how definitions from shared libraries, when used, are represented in measure logic.

Example of a CQL definition⁵ using the local shared library, **HospiceQDM**:

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

³ See the global shared library, CQMCommonQDM v9.0, at <https://ecqi.healthit.gov/sites/default/files/2027-CQMCommonQDM-Library.zip>.

⁴ Example from CMS130v14 for 2026 reporting.

⁵ Example from CMS130v14 for 2026 reporting.

A. Best practices for naming primary CQL libraries

The maximum allowable length for a measure’s primary CQL library name is 64 characters, but the recommended length is 30 characters or less. The library name in the ‘Measure CQL Library Name’ field in MADiE should adhere to the following standards and should **not** contain any special characters⁶:

- eCQM Abbreviated Title + Shortened Measure Name

Example: CMS506SafeUseofOpioids

In MADiE, the ‘CMS ID’ field contains the numerical identifier for the measure (that is, 506), while the ‘eCQM Abbreviated Title’ field contains the prefix of “CMS” along with the measure’s numerical identifier (that is, CMS506). The ‘Measure Name’ field contains the full title of the eCQM (that is, Safe Use of Opioids - Concurrent Prescribing).

B. Best practices for naming shared CQL libraries

When naming a new shared CQL library, adhere to the following standards:

- **Do** capitalize the first letter of every word and remove all whitespace between words (PascalCase).⁷

Example: AdultOutpatientEncountersQDM

- **Do** use names that are short, descriptive, and easy to read and that accurately reflect the contents of the shared library. There is a 64-character maximum for shared library names; however, the recommended length is 30 characters or less.

Example: HospiceQDM

- **Do** append “QDM” to the names of shared libraries used in eCQMs.

Example: AdultOutpatientEncountersQDM

- **Do not** use special characters in the library name. The following is an example to avoid:

Adult+PediatricEncounter

- **Do not** start the library name with number. The following is an example to avoid:

2AdultEncounters

⁶ Special characters are symbols such as a plus sign (+), an underscore (_), a minus sign (-), an asterisk (*), and a slash mark (/). This list is non-exhaustive.

⁷ See Appendix B for a complete list of case type definitions.

4. STANDARDS: DEFINITIONS

Definitions are named CQL expressions that implement measure concepts and can be referenced by other definitions. Measure developers should reference other CQL definitions whenever appropriate.

A. Best practices for writing CQL definitions

When writing definitions, use the following best practices:⁸

- **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 or when you would like to return data from multiple sources together in a tuple or array.

Example of CQL definition using ‘from’:¹⁰

▲ Encounter with Antibiotic Ordered Within Three Days

```
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"
```

- **Do** use the same numeric thresholds and timing units in the logic as described in the measure header narrative. For example, if the measure header describes that a certain action must take place more than 180 minutes after the encounter start, then the associated logic should be represented with expressions using minutes, not hours.

⁸ For further information on best practices for writing CQL definitions, see the sections on Queries (<http://cql.hl7.org/02-authorsguide.html#queries>) and Full Query (<http://cql.hl7.org/02-authorsguide.html#full-query>) of the CQL Specification Author’s Guide.

⁹ Example from CMS146v14 for 2026 reporting.

¹⁰ Example from CMS2v15 for 2026 reporting.

As a reminder, the measure header, definitions, and logic should always be aligned.

B. Best practices for naming CQL definitions

When naming definitions, use the following best practices:

- **Do** capitalize the first letter of every word (initial case).¹¹

Examples of CQL definition names using initial case:¹²

▲ **Low Glucose Test Followed By Glucose Test Result Greater Than 80**

▲ **Encounter With Antibiotic Ordered Within Three Days**

- **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. Please note that long definition names can impair readability of the logic.

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**

- **Do** use only abbreviations or acronyms that are spelled out and used in the measure's header 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 or no return should be named like a question, using the words “is” or “has,” and definitions that return multiple encounters should indicate plurality.¹⁶

¹¹ See Appendix B for a complete list of case type definitions. The use of initial case for all CQL definition names will be required for the first time in the 2027 reporting/performance period eCQMs. This guide will continue to be updated with definition name examples that use initial case in future iterations.

¹² Examples from CMS816v5 and CMS146v14 for 2026 reporting.

¹³ Examples from CMS136v15, CMS1028v4, and CMS135v14 for 2026 reporting.

¹⁴ Examples from CMS108v14 and CMS144v14 for 2026 reporting.

¹⁵ MADiE will display the return type in the definition form (when there are no CQL errors).

¹⁶ For episode-based eCQMs, measure developers may exercise their discretion when naming definitions to indicate plurality of returns based on their understanding of the potential return of each logic definition.

Examples of CQL definition names:¹⁷

- ▲ Has Most Recent Adult Screening Negative
- ▲ Has Total Hip Arthroplasty With 1 Or More Lower Body Fractures
- ▲ Has Normal BMI
- ▲ Is Age 66 Or Older Living Long Term In A Nursing Home

- **Do not** give a definition the same name as a value set, 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:

▲ **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.
- **Do not** use special characters¹⁹ in definition names. The following is an example to avoid:

% 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

¹⁷ Examples from CMS2v15, CMS56v14, CMS69v14, and CMS125v14 for 2026 reporting.

¹⁸ See the CQL Specification Author’s Guide for a full list of operators: <https://cql.hl7.org/STU4/02-authorsguide.html#operations>.

¹⁹ Special characters are symbols such as a plus sign (+), an underscore (_), a minus sign (-), an asterisk (*), and a slash mark (/). This list is non-exhaustive.

Below are more examples of CQL definition names 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"
  ThirtySevenWeeksPlusEncounter
  with ["Procedure, Performed": "Cesarean Birth"] CSection
  such that Global."NormalizeInterval" ( CSection.relevantDatetime, CSection.relevantPeriod )
  during PCMaternal."HospitalizationWithEDOBTriageObservation" ( ThirtySevenWeeksPlusEncounter
  )
```

Example of CQL definition:²¹

▲ **Most Recent Glycemic Status Date**

```
Last(("Glycemic Status Assessment" QualifyingGlycemicStatus
  return date from start of Global."NormalizeInterval"(QualifyingGlycemicStatus.relevantDatetime,
  QualifyingGlycemicStatus.relevantPeriod)) QualifyingGlycemicStatusDays
  sort asc
  )
```

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 <i>or</i> Exclusions for Hospice	Has Hospice Services
Encounters <i>or</i> Eligible Encounters <i>or</i> Valid Encounters	Qualifying Encounters (Note: generally used for clinician measures)

²⁰ Example from CMS334v7 for 2026 reporting.

²¹ Example from CMS122v14 for 2026 reporting.

5. STANDARDS: ALIASES

Aliases 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, definition query sources, and let statements.

A. Best practices for using CQL aliases and argument names

When naming aliases, apply 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 starts before or on day of start of
          Global."NormalizeInterval" ( HighInterventionsPerformed.relevantDatetime,
            HighInterventionsPerformed.relevantPeriod )
          and not ( OverweightObese.prevalencePeriod ends before day of start of
            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. ["Assessment, Not Performed": "Adolescent depression screening assessment"]
NoAdolescentScreen
 2. ["Intervention, Performed": "Tobacco Use Cessation Counseling"] TobaccoCessationCounseling
 3. ["Diagnosis": "Prostate Cancer"] ActiveProstateCancer
 4. ["Diagnosis": "History of Atrial Ablation"] AtrialAblationDiagnosis
- **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 CMS69v14 for 2026 reporting.

²⁴ Examples from CMS2v15, CMS138v14, CMS129v15, and CMS71v15 for 2026 reporting.

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

Examples of CQL aliases:²⁵

1. EncounterLastDiastolicBP: Last(["Physical Exam, Performed": "Diastolic blood pressure"] **DiastolicBP**)
2. ["Medication, Order": " Beta Blocker Therapy for LVSD"] **BetaBlockerOrdered**
3. "Documented Low BMI During Measurement Period" **LowBMI**

- **Do** use only abbreviations or acronyms that are spelled out and used in the measure’s header narrative.

Examples of CQL aliases (Note: In each example, the acronym is defined in the narrative.):²⁶

1. ["Assessment, Performed": "VR-12 Physical component summary (PCS) score – oblique method T. score"] **VR12PhysicalAssessment**
2. "No VTE Prophylaxis Medication Administered Or Ordered" **NoVTEMedication**

- **Do not** reuse the same alias for different data elements within the same measure, because 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 to represent a distinct data element.

```
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, because this can reduce the clarity and readability of the logic.
- **Do not** use an alias if the definition statement does not require additional logic, because 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.

²⁵ Examples from CMS22v14, CMS144v14, and CMS69v14 for 2026 reporting.

²⁶ Examples from CMS56v14 and CMS108v14 for 2026 reporting.

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 or 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. If referring to one specific type of encounter, use the specific encounter type in the alias name, as illustrated below.) InpatientEncounter HeartFailureEncounter PsychotherapyEncounter
["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 <i>and</i> Heart failure diagnosis	HeartFailureEncounter <i>and</i> HeartFailureDiagnosis

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 pre-defined functions from the global shared library. Functions perform operations on input arguments, while definitions operate only on the expressions in the definition.

A. Selecting functions

Measure developers are encouraged to select the function that is most appropriate to meet the measure’s intent from the pre-defined list of functions from the global shared library. The following are examples of preferred functions from the global shared library.

- **Global."NormalizeInterval"()** function:²⁷

▲ **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.²⁸
- The **NormalizeInterval** function might be needed when using a sort clause.

Example of NormalizeInterval function with sort clause (used in logic):²⁹

▲ **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) during
day of QualifyingEncounter.relevantPeriod
sort by start of Global."NormalizeInterval"(relevantDatetime, relevantPeriod)
),
```

- Age functions:

Example of CQL age function: "AgeInYearsAt"()

"AgeInYearsAt" (date from start of)

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

²⁷ For more information on the NormalizeInterval function, please review the “Teach Me CQL: Normalize Interval” video at https://www.youtube.com/watch?v=PKKta_fKKh8.

²⁸ Review the QDM documentation on the eCQI Resource Center for a full list of QDM datatypes and their attributes: <https://ecqi.healthit.gov/qdm>.

²⁹ Example from CMS22v14 for 2026 reporting.

Example of AgeInYearsAt function (used in logic):³⁰

▲ **Initial Population**

"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 length-of-stay function: Global."LengthInDays"()

- The **LengthInDays** function calculates the difference in midnight-to-midnight 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.

Example of LengthInDays function (used in logic):³¹

▲ **Encounter with First ICU Location Stay Less Than 1 Day**

"Encounter With ICU Location" QualifyingEncounterICU
where Global. "LengthInDays" ("FirstICULocationPeriod" (QualifyingEncounterICU)) < 1

Example of CQL length-of-stay function:

Global."HospitalizationWithObservationLengthOfStay"()

- This function 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 inpatient encounter.

▲ **Global.HospitalizationWithObservationLengthofStay(Encounter "Encounter, Performed")**

"LengthInDays"("HospitalizationWithObservation"(Encounter))

Example of HospitalizationWithObservationLengthOfStay function (used in logic):³²

▲ **Encounter Less Than Two Days**

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

Example of CQL length-of-stay function:³³

Global."HospitalizationWithObservationAndOutpatientSurgeryService"()

- This function returns the total interval from the start of any immediately prior emergency department visit, outpatient surgery visit, or observation visit—when the transition between discharge from these encounters and admission to the inpatient encounter is one hour or less—to the discharge of the given inpatient encounter.

³⁰ Example from CMS177v14 for 2026 reporting.

³¹ Example from CMS190v14 for 2026 reporting.

³² Example from CMS72v14 for 2026 reporting.

³³ Example from CMS529v6 for 2026 reporting.

▲ **Global.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]
```

- Example of CQL length-of-stay function:³⁴ **Global."HospitalizationWithObservation"()**

This function returns the total interval from the start of any immediately prior emergency department visit or observation visit—when the transition between discharge from these encounters and admission to the inpatient encounter is one hour or less—to the discharge of the given inpatient 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]
```

³⁴ Example from CMS71v15 for 2026 reporting.

B. Best practices and standards for naming CQL functions

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.³⁵

Example of CQL function name:³⁶

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 pre-defined function, because this can raise the risk of logical errors.

Process step: Please review the pre-defined functions available in MADiE to ensure the name of your function is not the same as that of any of the pre-defined functions.

³⁵ See Appendix B for complete definitions of case types.

³⁶ Example from CMS334v7 for 2026 reporting.

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"

For hospital measures, the recommended timing pattern for initial population criteria is an encounter that **ends during day of "Measurement Period."** The use of **"ends"** provides a single point of time comparison within the measurement period so the receiving systems can provide reports on a quarterly basis, while **"day of"** specifies day precision to avoid time zone offset and millisecond issues.

Example CQL from hospital eCQM:³⁷

▲ Encounter with Age 18 and Older

```
["Encounter, Performed": "Encounter Inpatient"] InpatientEncounter  
  where InpatientEncounter.relevantPeriod ends during day of "Measurement Period"  
  and AgeInYearsAt(date from start of InpatientEncounter.relevantPeriod) >= 18
```

For eligible clinician measures that use encounters in the initial population, the recommended timing pattern for initial population criteria is for encounters **during day of "Measurement Period."**

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

Example CQL:³⁸ Initial Population = Denominator

▲ Initial Population

```
"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 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

³⁷ Example from CMS826v3 for 2026 reporting.

³⁸ Example from CMS645v9 for 2026 reporting.

mathematics notation, inclusive (closed) boundaries are indicated with square brackets, and exclusive (open) boundaries are indicated with parentheses.”³⁹

- Consider the following statement: Interval[3, 5]

This expression results in an interval that contains the integers 3, 4, and 5, because the starting and ending brackets represent closed or inclusive boundaries, meaning that the boundary points are included in the interval.

- Consider the following statement: Interval(3, 5)

This expression results in an interval that contains only the integer 4, because the starting and ending parentheses represent open or exclusive boundaries, so 3 and 5 are not contained in the interval.

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:⁴⁰ Age Interval

▲ Initial Population

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

- **Do not** use mixed-boundary notation for an interval in the same logical statement. See the following for an example to avoid:

Interval(18, 85]

B. Additional timing phrases

CQL supports precision-based date and time comparisons. Be sure to consider whether date or dateTime is being evaluated in timing phrases.⁴¹ 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.

³⁹ See the CQL Specification Author’s Guide for more on interval values: <https://cql.hl7.org/02-authorsguide.html#interval-values>.

⁴⁰ Example from CMS165v14 for 2026 reporting.

⁴¹ See the CQL Formatting and Usage Wiki for more on timing phrases: (1) <https://github.com/cqframework/CQL-Formatting-and-Usage-Wiki/wiki/Authoring-Measures-in-CQL#time-interval-calculations> and (2) <https://github.com/cqframework/CQL-Formatting-and-Usage-Wiki/wiki/Cooking-with-CQL-Q&A:---Date-and-Time-Calculations>.

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

Example CQL:⁴² "ends 1 day after day of start of"

A supplementary timing constraint is added to ensure that the timing of the relevant period for the **AnesthesiaProcedure** ends one day after the start of the 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
```

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 assess 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.⁴³

▲ TJC.Non Elective Encounter With Age

```
["Encounter, Performed": "Nonelective Inpatient Encounter"] NonElectiveEncounter
where AgeInYearsAt(date from start of NonElectiveEncounter.relevantPeriod) >= 18
and NonElectiveEncounter.relevantPeriod ends during day of "Measurement Period"
```

C. Operator precedence

CQL uses in-fix operator notation. When multiple operators with the same precedence appear in an expression, they are evaluated from 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 Appendix Table A.1 for more details.

CQL adopts an operator precedence to ensure consistent and predictable behavior of written expressions. For example, primary operators—such as, [] and ()—have the highest operator precedence, while conjunctions and disjunctions are lower on the list.⁴⁴ As with most expression

⁴² Example from CMS108v14 for 2026 reporting.

⁴³ Example from CMS71v15 for 2026 reporting.

⁴⁴ See Appendix Table A.1 for the full list.

languages, parentheses can always be used to force the order of operations if the defined operator precedence does not produce the intended evaluation of a given expression.

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:⁴⁵ 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. Value sets

Measure developers maintain and publish value sets on the [Value Set Authority Center](#) website.⁴⁶ Value sets and codes are listed in the terminology section of each measure specification.⁴⁷ Value sets are referenced in the logic in two equivalent ways as demonstrated in the examples⁴⁸ below.

Example: ‘In’ operator

▲ Delivery Encounter With Gestational Age Greater Than Or Equal To 37 Weeks Based on Coding

```
PCMaternal."Delivery Encounter With Age Range" DeliveryEncounter  
let CGA: PCMaternal."CalculatedGestationalAge" ( DeliveryEncounter ),  
    EGA: PCMaternal."LastEstimatedGestationalAge" ( DeliveryEncounter )  
where CGA is null  
and EGA is null  
and exists ( DeliveryEncounter.diagnoses EncounterDiagnoses  
             where EncounterDiagnoses.code in "37 to 42 Plus Weeks Gestation"  
           )
```

⁴⁵ Example from CMS125v14 for 2026 reporting.

⁴⁶ The versions of value sets and direct reference codes contained in eCQMs for each reporting/performance year can be accessed on the VSAC Downloadable Resources page: <https://vsac.nlm.nih.gov/download/ecqm>.

⁴⁷ For more information on value sets, please review the eCQI Resource Center’s Value Set Information page: https://ecqi.healthit.gov/value-set-information?qt-tabs_vsg=about.

⁴⁸ Examples from CMS334v7 for 2026 reporting.

Example: Shorthand

▲ Delivery Encounter With Cesarean Birth

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

E. Direct reference codes

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 codes and may be used for other single-use terminology codes.⁴⁹ 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 file; it might also be referenced in the Data Criteria section if used as part of a QDM data element, not just as an attribute of a previously defined QDM data element.

Example:⁵⁰ 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)")
```

F. Other style considerations

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

- **Do not** copy and paste auto-formatted content from a Microsoft Word Document into measure header text in MADiE. **Auto-formatted content (for example, straight quotes that have been changed to smart quotes, double hyphens that have been changed to em dashes, bulleted lists) when pasted into the measure header text in MADiE can cause issues with exporting or the export artifacts.** Please note, rich text formatting, including bold, italics, tables and bulleting, is now available for use in some fields directly in the MADiE tool.
- **Do** use the equivalence operator (~) to indicate equivalence between two values. Use the inequivalence operator (!~) to indicate inequivalence between two values. Equivalence is a semantic comparison, whereas equality requires a stricter, exact comparison.

⁴⁹ For more information on code systems, vocabularies, and terminologies, please review the MMS Hub's Specify the Code resource: <https://mmshub.cms.gov/measure-lifecycle/measure-specification/specify-code/code-systems-vocabularies-terminologies>.

⁵⁰ Example from CMS117v14 for 2026 reporting.

Example:⁵¹

▲ Qualifying Encounter During Day of 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 day of "Measurement Period"
  and QualifyingEncounter.class !~ "virtual"
```

Example:⁵²

```
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)"
  )
```

- Do use the equality operator (=) or inequality operator (!=) to identify exact matches between two values.

Example:⁵³

▲ Increase of 0.3 or More Using Lowest Creatinine within 24 Hours

```
from
  "Encounter with Creatinine and without Obstetrical Conditions" QualifyingEncounter,
  ["Laboratory Test, Performed": "Creatinine Mass Per Volume"] IndexCreatinineLabResult,
  ["Laboratory Test, Performed": "Creatinine Mass Per Volume"] SubsequentCreatinineLabResult
  let IndexCreatinineLabResultTime: Global."EarliestOf" (
  IndexCreatinineLabResult.relevantDatetime, IndexCreatinineLabResult.relevantPeriod ),
  SubsequentCreatinineLabResultTime: Global."EarliestOf" (
  SubsequentCreatinineLabResult.relevantDatetime, SubsequentCreatinineLabResult.relevantPeriod ),
  HospitalWithObservation: Global."HospitalizationWithObservation" ( QualifyingEncounter )
  where ( SubsequentCreatinineLabResult.result.value ) - ( IndexCreatinineLabResult.result.value )
  > 0.299
  and IndexCreatinineLabResult.result.value == "LowestSerumCreatinine"(QualifyingEncounter)
  and IndexCreatinineLabResultTime during Interval[start of HospitalWithObservation,
  SubsequentCreatinineLabResultTime]
  and IndexCreatinineLabResultTime during HospitalWithObservation
  and IndexCreatinineLabResultTime during Interval[start of HospitalWithObservation, start of
  HospitalWithObservation + 24 hours]
  and SubsequentCreatinineLabResultTime during HospitalWithObservation
  and SubsequentCreatinineLabResultTime during Interval[start of HospitalWithObservation,
  start of HospitalWithObservation + 48 hours]
  and IndexCreatinineLabResult.id != SubsequentCreatinineLabResult.id
  return QualifyingEncounter
```

⁵¹ Example from CMS142v14 for 2026 reporting.

⁵² Example from CMS146v14 for 2026 reporting.

⁵³ Example from CMS832v3 for 2026 reporting.

VERSION HISTORY

Version	Date	Description of change
2.0	August 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
4.0	May 2020	Updated examples using birth date to reflect the addition of birth date, a direct reference code
		Updated language and content to align with standards changes QDM 5.5 and CQL 1.4
5.0	May 2021	Updated examples of logic to reflect most recent use
		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
9.0	May 2025	Updated examples of logic to reflect most recent use
		Updated all tooling references from MAT to MADiE
		Added new examples, references, and guidance on case types
		Updated title of guide from 'CQL Style Guide' to 'QDM-based CQL Style Guide'
10.0	May 2026	Updated examples of logic to reflect most recent use
		Added new examples, references, and guidance on case types

APPENDIX A:

THE ORDER OF OPERATOR PRECEDENCE IN CQL

The Order of Operator Precedence in CQL

Operator precedence determines the order in which operators are evaluated in expressions. Parentheses can be used to override the default precedence and ensure expressions are evaluated in the desired order.

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” <i>component</i> “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” <i>precision</i> “between” <i>“duration in” precision</i> “between” <i>“difference in” precision</i> “between”
Binary List	“union intersect except”
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”

Source: CQL Specification Developer's Guide, Operator Precedence Section (<https://cql.hl7.org/N1A/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 white spaces between 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 white spaces between characters allowed.
- **Title Case.** Standard title casing, including spaces and tabs, but no other white spaces between characters allowed.
- **Initial Case.** First letter of every word is capitalized (for example, “Includes Or Starts During”), as opposed to title case, which traditionally does not capitalize conjunctions and prepositions (for example, “Includes or Starts During”).