Pioneers in Quality Electronic Clinical Quality Measure (eCQM) Clinical Quality Language (CQL) Basics Webinar for Hospitals

November 29, 2018

Tricia Elliott: Hello, everyone. I'm Tricia Elliott, the Director of Quality Measurement in the Department of Quality Measurement at the Joint Commission. I would like to thank you for joining us for, "Pioneers in Quality: Electronic Clinical Quality Measure (eCQM) Clinical Quality Language (CQL) Basics Webinar for Hospitals."

Both CMS and the Joint Commission will move from the previous quality data model logic expression language to CQL for the 2019 reporting period. As a result of the feedback received from stakeholders interviewed through the CMS eCQM Strategy Project to reduce implementation and reporting burden, this session will provide an introduction regarding CQL expression language. This session is intended to prepare hospitals for this transition and serves as a precursor to a more focused technical educational webinar series that starts in December 2018 and continues through March 2019 that will offer a deeper dive into the specifics of CQL logic for each individual eCQM. Each one-hour expert-to-expert session will be dedicated to discussing two to three eCQMs and the new CQL logic expression language.

After today's webinar, you will be able to describe basic concepts about the new eCQM CQL expression language logic. You will be able to prepare for implementation of the CQL logic expression language for the 2019 eCQM reporting period. In addition, you'll be able to locate additional resources and where to ask questions about eCQMs and CQL.

These speakers disclosed that neither they nor their spouses or partners have any financial arrangements or affiliations with corporate organizations that either provide educational grants to this program or may be referenced in this activity.

Shanna Hartman, MS, RN, nurse, Division of Electronic and Clinical Quality Center for Clinical Standards and Quality with the Centers for Medicare and Medicaid Services, and Bryn Rhodes, Standards Support SME for ESAC, thank you for joining us today.

Shanna Hartman: Hi, everyone. This is Shanna Harman from CMS, and we thank you for joining today's "eCQM Clinical Quality Language Basics for Hospitals." I'll be presenting today, along with Bryn Rhodes from ESAC, Incorporated. Some of the quality language, or CQL, is a Health Level Seven (HL7) International standard designed to unify the expression of logic for electronic clinical quality measures (eCQMs) and clinical decision support (CDS). CQL provides the ability to better express logic-defining measure populations to improve the accuracy and clarity of eCQMs.

There are many benefits of CQL...such as improved expressivity, more precise and unambiguous, can share logic between measures, can share logic with decision support, can be used with multiple information data models, and simplifies calculation engine implementation.

eCQMs were transitioned to use the Clinical Quality Language Standard for logic expression beginning with calendar year 2019 reporting period for hospitals. The following programs for hospitals of CMS are impacted by this transition:
Hospital inpatient quality reporting programs

Medicare and Medicaid Promoting Interoperability Programs, previously known as the Electronic Health Records Incentive Program.

This also is applicable to eligible professionals and eligible clinician eCQMs in the following programs...the Medicare and Medicaid Promoting Interoperability Program, as well as the Quality Payment Program, Merit-based Incentive Payment System, and Advanced Alternative Payment models such as Comprehensive Primary Care Plus. To support this transition, CMS published CQL-based eCQMs in the spring of 2018.

At this time, I will hand the presentation over to Bryn to go into more about what is CQL.

Bryn Rhodes: I'm excited for the opportunity to present on this topic. I've been part of the development of clinical quality language for several years now. It's exciting to see it starting to be used to share quality measurement specifications. During this presentation, we will look at lots of examples of CQL as it appears in the EH and EAH quality measures. But first, I'd like to start by answering this question: What is CQL?

For the answer to this question, I'd like to take a step back and ask another question...what is a quality measure?

A tool to assess performance related to a specific clinical process, structure, or outcome...this definition comes from Health Quality Measures Format, the HQMS normative specification. So when we think about what a quality measure is and how to describe it, one of the things we're most interested in, especially in terms of performing that measurement over a broad range of systems and environments, is communicating the definition of the quality measure.

One way that communication happens is through narrative descriptions that explain in natural language what is being measured, what patient or other clinical information is involved, what the various criteria are for inclusion or exclusion, and how the actual calculation of the metrics should be performed. The narrative descriptions like this of the quality measures are good, but electronic representations allow for easier implementation in the hospital systems.

An electronic clinical quality measure is then a digital representation with the goal of enabling the measure to be evaluated automatically, so this allows us to more precisely describe the intent both in terms of the data involved and the relationships and criteria in the metrics. So in fact, eCQMs carry both the complete narrative description as well as more formal machine-readable representation.

For example, let's take a look at a snippet of narratives from CMS 104, "Discharged on Antithrombotic Therapy." This is the overall description of the measure of ischemic stroke patients prescribed or continuing to take antithrombotic therapy at hospital discharge. So that's a nice distinct description of the quality measure, but there's a lot of information packed into that; and there are a lot of questions about that statement that arise in trying to communicate that definition...Who said it? When did they say it? What evidence supports it? How should I use it? These are all questions about this description. We categorize those as metadata about the measure.

Then there are questions about the content of the description. What kind of *things* does it talk about...patient, ischemic stroke, prescriptions, just the hospital discharge? What do those things look like? What kind of attributes do they have? What kind of facets does that information carry? That we categorize as the data models, the structure of things that we're talking about. Then what are the relationships between them and what are the criteria that apply to them...so things like continuing to take and therapy at hospital discharge? Those are relationships, often temporal, between the data involved; and those we characterize as the logic involved.

If we look at eCQM representation and the specifications that are used to do that, health quality measures format and quality data model are the specifications that have been used up to this point; and across those three categories of information, metadata is carried in health quality measures format. So that carries through the author and what the evidence is and citations and the narrative description of the structure of the measure.

The data model and the logic up to this point were both contained within quality data models, so that describes the type of data we're interested in; and it involves the relationships between those data elements. The new specifications...we've taken the logic portions of the quality data model, separated them into a separate specification, and added functionality to make more sophisticated quality measures possible to express.

So looking specifically at quality data model, it's a conceptual model that provides a way to formally specify clinical statements. QDM starts at the highest level with categories of information...which is laboratory tests, diagnostic studies, encounters, and so on. QDM then defines different contexts for those categories...things like performed, ordered, or dispensed. The categories and contexts are combined to let us talk about things like medication administered or encounter performed.

From there, we combined that to a particular terminology to let us say exactly what kind of information we're talking about. For example, we can use RxNorm terminology to indicate that we want antithrombotic therapy, the medication that was prescribed. Note that we'll speak more about terminology a little later in the presentation; but finally with QDM, for each data element we can talk about the attributes of the individual pieces of information that are part of that data element.

For example, in an encounter performed, these are the types of information that we need or associate with an encounter...so relevant period, discharge disposition, principal diagnosis. Note that the relevant period attribute replaces admission time and discharge time; and the relevant period is an interval, so it has a start and a stop. We'll see more about interval operations later in the presentation. Note also the Id attribute lets you talk about specific instances of data elements, and the code attribute allows you to talk about the terminology for a specific data element.

Then one more thing to note...if you look at diagnoses, that's actually a plural element. So if there are multiple diagnoses reported with the encounter, you'll see references to that and the ability to talk about all of the diagnoses associated with the encounter.

Then when we talk about what it takes to share the logic or the criteria involved in the specifications of a measure, we separate the description of the terminologies and the description of the data model from the description of the logic. Because of this separation, we can isolate the impact of changes on the specification. For example, we can introduce new operations into the logic without having to change quality data models. We can change quality data model structures without having to change the logic specification.

In addition, just like QDM can reference different terminologies, as a separate specification CQL can now reference different data models as well as different terminologies. So then coming back to this question...what is CQL...it's a language that we can use to share precise machine-readable and human-readable measure definition. One more thing to note...although we've described CQL from the perspective of quality measurement, we can also use it to describe the logic involved in distributing decision support; and we can share complementary decision support that uses this same logic between measures and the decision support.

Okay, so with that background for what CQL is, now we'll switch gears and take a deeper dive into what CQL looks like, how it's being used in the EH and CAH eCQMs. Many of the examples we'll look at come from the same measure, CMS 104; but we'll look at some others as well to give you a sense for all the different constructs you'll encounter in the CQL EH and CAH measures.

So this is the population criteria section of CMS 104, "Discharged on Antithrombotic Therapy." There's a proportion measure, so we see the typical criteria sections...the initial population, numerator, denominator, et cetera. Two things to call out here...first, the initial population, this is referring to a single definition with a meaningful name, encounter with principal diagnosis on age. With CQL, measure authors can use definitions to build up population criteria; and they can give those definitions names to capture the meaning of the expressions they represent.

Second, note the use of this "TJC.". That's a reference to a library called "TJC." With CQL, measures can reuse definitions across measures by packaging them in a library. So instead of repeating the definition of "encounter with principal diagnosis and age" in the measures that use it, it can be defined once and then shared between any number of measures. So CQL uses definitions like building blocks, using one definition to build upon another and another and another.

Using very simple terms, this diagram depicts a visual representation of a generic patient population and its structure. Each block represents a definition in the CQL. Each definition addresses a specific population parameter. So the base of the diagram is the largest block with the largest population, as indicated by the icon; that one is "A" for encounters. In this piece of logic, we're looking for a specific encounter type; for example, non-elective encounters.

Moving up to block "B," we don't need to repeat the encounter logic; simply call in definition "A" and add logic for diagnosis. So then definition "B" is "A" and diagnosis, such as ischemic or hemorrhagic stroke.

Then moving up to block "C," we add in definitions "B" and "A's" logic. So note the pattern where we build off each definition by adding additional criteria. Each successive definition further constrains the logic, narrowing down the patient population. The "C" at the top levels encompasses all criteria through definitions "A" and "B" and becomes the initial population definition.

In case you're wondering, "Why not just write the logic where "C" equals encounters and age and diagnosis all in one statement," it's a fair question; but we follow the methodology, using lots of definitions because it's cleaner to read and allows us to reuse definitions throughout the measure specifications without having to rewrite each piece of logic. It also gives the opportunity to communicate the intent of the expression through the name used for the definition.

So making this concrete, looking at "A" here, CMS 104 starts with non-elective inpatient encounter; and then it adds the diagnosis criteria in all stroke encounters and then adds age criteria in encounter principal diagnostics and age. Then as we saw in the population criteria, the measure just uses encounter with principal diagnosis and age as its initial population.

Now let's look at the anatomy of the definitions, as you'll see it in the human-readable. The definitions in CQL have a name, and that's an identifier that's unique within the library and an expression with content of the definition. As we've seen, these expressions can reference other definitions to build up more complete expression.

Digging into expressions, expressions in CQL are made up of values...like numbers and quantities; operators, like less than or equal to, plus, or and during; and identifiers, like non-elective inpatient encounters. CQL allows these components to be combined according to the rules of the language, such as you can add two numbers together to get another number so that they'll always produce a result. For example, "two plus two" is an expression; and the result is four. In this case, we're retrieving encounters that have a code in the non-elective inpatient encounter value set, that have a length in days of the relevant period equals the 120, and that ends during the measurement period.

Let's dig in further to that. Queries are an important kind of expression that you'll see operating on lists of values. But like lists of encounters or lists of lab orders, this expression is actually a query. You can tell because query starts by introducing the source and giving it an alias. An alias is an identifier that's used in the query to refer to the items in the source. That alias then is used in clauses like the "where" clause to build up (inaudible) criteria.

Digging in further, the source of that query is a special kind of expression in CQL called a "Retrieve." This is where we're specifying what data we're actually looking for in the EHR. Retrieves are also enclosed in square brackets and have a type and a terminology. Retrieve expressions are effectively the same as data criteria in previous versions of CDM in that they specify the data of interest in terms of the CDM category and data type, as well as the value set that specifies what codes should be returned.

Note that the terminology is typically a value set, but it may also be a code system or even a single direct reference code. Thus, Retrieve is saying, "Find all the completed encounters with a code in the non-elective inpatient encounter value set," and that will be the result in this expression...that list of encounters that match that criteria.

As an aside, let's look at the Terminology section for this measure. Within HQMS, now you'll see a Terminology section that references or defines all of the terminology that's used within that measure. The names given to the value sets are all available throughout the measure. The Retrieve we just saw is referenced in the non-elective inpatient encounter. If a measure uses any code systems or direct-reference codes, they'll also be listed in the terminology section; and these value sets are all available in the value set authority center. For example, searching for the antithrombotic therapy value set by its object identifier, or OID, you can see the value set displayed here. The contents of the value set can also be viewed from here.

Let's turn back to the Retrieve; let's look at another example. For some QDM data elements, a negated form is defined. That means the event or data element is not present. So this allows us to look for events that didn't happen; for example, a patient was not discharged with a particular medication. For all negated types in QDMs, the attribute "negation rationale" is used to provide the reason for the absence of why the event didn't occur. In this case, the expression returns "Antithrombotic therapy medications that were not discharged," having the rationale of either "Medical Reason" or "Patient Refusal."

Note the reference to terminologies using the IN operator for value set membership testing. So this operation can be read into negation rationale codes in the "Medical Reason" or "Patient Refusal" value set.

Turning back to the value set authority center, terminologies used in the hospital measures can all be downloaded in various formats, including Excel spreadsheets, by use the "Download" tab plus the appropriate reporting and performance period and select the format and sort order...Excel and CMS IDs in this case. What you get is an Excel spreadsheet looking at the specific entries in that spreadsheet for this measure. It contains all the value sets and their contents used. Note that for the "Medical Reason" and "Patient Refusal" value sets, the criteria is looking for the attribute negation rationale; so they don't have an associated QDM category.

In prior versions of QDM, this QDM category listing actually contains the attribute reference. Now the QDM category will only be specified for the terminology when that terminology is referenced from a Retrieve...so it's actually part of the data criteria of the measure.

Let's look a little deeper at direct-reference codes. CQL measure specifications can now use direct-reference codes...meaning that instead of a value set, it contains a list of codes. Criteria can be specified to reference a code directly. So just as with value sets, these codes can be made up and make up part of the data criteria if you're inside the Retrieve, such as this example, "Assessment, Performed" "Risk for Venous Thromboembolism," or they can appear anywhere outside a Retrieve as part of a CQL expression. In the human-readable Terminology section, the code and code system are specified...in this case, the code system and link, Version 2.63; and the code is "Risk for Venous Thromboembolism." In this case, because the terminology reference is within a Retrieve, this usage also appears in the Data Criteria section.

The second example taken from CMS32, "Median Time from ED Arrival to ED Departure for Discharged ED patients," the comparison tests whether the discharge disposition attribute is equivalent to the SNOMEDCT PT code "Patient

Deceased During Stay." As with value sets, because this terminology is referenced outside of a Retrieve, it appears in the Terminology section but not in the Data Criteria.

One more thing to note about the direct-reference code comparisons, you'll see it using a tilde rather than an equal sign. CQL allows both equality and equivalence comparisons. Equality means they're exactly the same value. Equivalence allows for the same meaning but not necessarily the same value per code specifically. This means the same code system and code, but not necessarily the same code system version or display. For example, if the discharge disposition was coded with the same SNOMEDCT PT code but from a previous code system version, the codes are equivalent but not equal. That's important for when we need to update code systems after the measure specifications have been released.

Just as the value sets in the value set authority one more time, the information required to communicate value sets and codes is different. The direct-reference codes used by eCQM specifications are provided in a separate download. Again, on the "Downloads" tab, you'll see a new entry. The last row in the table lists direct-reference codes specified within eCQM HQMS files. So this file contains all the direct-reference codes used in both hospitals and clinician eCQMs; and if you download that, you'll get an Excel spreadsheet that contains all the codes, the code systems, and what eCQM is used in.

So turning back to the "where" clause in this query, the "where" keyword introduces the clause; and what follows is an expression that results in a "Yes" or "No"...indicating whether each element and its source should be included in the results. In this case, for each encounter in the source the length in days of the encounter is less than or equal to 120; the relevant period is the encounter ends during the measurement period. The result is "Yes," and the encounter is included in the results.

So recall that the name "Non-elective Encounter" was introduced as an alias in the query for the encounter elements in the source. So for data elements, the dot is used to access the attributes of the data elements. The non-elective encounter dot relevant period is accessing the relevant period attribute that the encounters in the source named "Non-elective Encounter."

For "Encounter Performed," the relevant period is referring to the period between admission and discharge. Note that the attribute name "Relevant Period" is used across QDM on different types and can generally be thought of as the clinically-relevant period. Using consistent names for attributes across types makes it easier to learn and understand the conceptual data model defined by QDM.

There are two other things to call out here. First, notice the use of the Function's "LengthInDays." The Function is defined in the Global Library, the common library that's used to share definitions and functions among all the program measures.

Second, note the Attribute "Relevant Period" and the "Parameter Measurement Period" are both intervals, meaning they can have a starting point and an ending point. CQL defines operations for comparing and evaluating intervals directly, as well as timing phrases that allow temporal relationships to be expressed with natural language such as "and during." So because timing relationships are such an important part of describing criteria and quality measures, CQL supports a broad range of operations to compare dates and times and intervals of data.

For example, you can compare two daytime values...such as Example 1 here, which can be read, "The encounter was documented before the assessment was completed." Or you compare date/time with an interval, such as Example 2. The assessment was completed during the encounter. We can also compare an interval with a date/time, such as Example 3. The encounter admission to discharge time includes the assessment completion. Now, that's just the converse of the previous example. Finally, we can compare two intervals directly...the encounter took place during the measurement period.

Looking at a few more examples of the types of timing phrases CQL supports, we can just compare intervals...so other interval operators like overlaps, meets, before, after. Example 1 here shows the use of overlap, "Prevalence period of the diagnosis indicating elective delivery overlaps the delivery encounter."

The second example, timing phrases can use the keywords "starts" and "ends," access the beginning or ending of an interval, similar to the way QDM specified timing relationships such as "starts before start" and "during." All can be used.

Timing phrases can also include offsets, such as "The labor was documented 24 hours or less before the start of a C-section encounter" or, "The Emergency Department visit with an admit order and one hour or less before or on the start of the encounter."

Timing phrases, lastly, can also specify the precision at which the comparison is performed...so to specify that the comparison should be performed only to the date, ignoring the time component. The "day of" keywords are used in the timing phrases. In this example, the anesthesia procedure ends one day after the qualifying encounter admission.

Next we'll look at one of the most common concepts in CQL, the "Relationship" clause. So relationships in CQL use the "with" and "without" keywords to establish relationships between sources. From this example, the query describes a relationship between the Primary Source, ischemic stroke encounter, and the Related Source, antithrombotic therapy at discharge. The Related Source is given an alias as well...Discharge Antithrombotic...so you can talk about it within the condition in the "such that" to define the criteria for the relationship. So this example can be read, "The numerator consists of ischemic stroke encounters with antithrombotic therapy at discharge such that the discharge antithrombotic author day/time is during the ischemic stroke encounter's relevant period."

Queries can also include any number of relationships with each additional relationship further restricting the primary source, only those elements that have the established relationships. From this example taken from CMS9v7, exclusive breastmilk feeding, the definition "Newborn fed breastmilk only since birth," can be read "Single led birth encounter with gestational age 37 weeks or more, the breastmilk feeding that started during the encounter's relevant period without dietary intake other than breastmilk that started during the encounter's relevant period."

Another example...sometimes we need to consider alternative relationships among the same source...relationships in the same query always adds to

"Restriction." So you'll see relative alternatives with relationships expressed by combining the results of each alternative using the (inaudible) operator, as seen in this example taken from CMS26v6, Home Management Planning Care, given the patient's caregiver. In this example, we are looking for either "Encounter with Discharge Disposition to Home or Police Custody with Asthma Plan Completed" or "No Asthma Management Plan due to Patient Refusal."

The alternatives are each expressed using a "with" relationship to the "Encounter with Discharge Disposition to Home or Police Custody." Those are then combined using (inaudible). Note the use of parentheses here. CQL allows parentheses to be used anywhere in the expression to indicate order of operation in the same way that parentheses can be used in any mathematical expression.

Another example...Multiple Sources, so "with" and "without" relationships are used when there are only two data elements in the relationship. There are cases where more than two elements need to be involved in order to properly describe the relationship. In those cases, CQL provides a multi-source query, as seen in this example taken from CMS113v7, Elective Delivery.

Since CQL multi-source queries begin with a "from" followed by any number of sources, in this query three different sources are introduced: "Delivery Encounter Near Term," "Medical Induction Medication," and "Is In Labor." The criteria for the relationship is then defined in the "Where" clause as labor was documented during the delivery encounter and the induction medication was administered 24 hours or less before the labor was documented. Notice the use of the "Return" keyword here. "Return" clause in CQL is used to shape the results of the query.

In this example because the query has three sources, each result may have up to three elements, one from each source. The "Return" is used here to indicate that the delivery encounter results should be included in the results of the query, so this will then return a list of delivery encounters that match this criteria.

Let's look more closely at the Union operator directly. Union is used to combine lists; it can combine any number of lists, and it eliminates duplicate as it does. As shown in this example taken from CMS102v7, Assessed for Rehabilitation, unions can combine lists of different types. This expression combines "Comfort Measures" interventions, "Ordered and Performed." The result will have elements of both types. So when different types are combined in a single list, different elements will have different attributes. For example, "Interventions Performed" have both a relevant period and an author date/time; but the "Interventions Ordered" only have an author date/time reference and either "relevant Period" attribute will not result in a value.

The Coalesce function is often used. The Coalesce function takes a list of arguments and returns the first one that results in a value. So in this case, it's the "Comfort Measures" and "Intervention, Performed" who had a relevant period; and that would be the result. Otherwise, the author date/time will be used.

Using Return to Shape Results...so in this example, when combining results of different types subsequent uses of that result will need to take into account that the list has different types of elements, so that can sometimes complicate (inaudible) expressions. In some cases, what you'll see is that the different types are treated as though they were the same, using the terms to shape the results of the query.

In this example, also taken from CMS108v7, the query combines risk assessments for being a thromboembolism with INR laboratory tests with a result greater than 3 in treating the latter as an assessment performed, using the result date/time and the "Laboratory Test" as the author date/time of the constructed assessment. The result of the expression is then a list of the substance performed elements. Some of those came from actual assessments performed in the underlying systems. Some of those were constructed in the results based on laboratory tests performed in the underlying systems.

The CQL also allows lists to be combined using "Intersect" and "Except." In this example, also from CMS 108, we see the use of Union and Intersect together. Notice the use of parentheses here to indicate order of operations? The innermost union of encounters with atrial fibrillation or hip or knee replacement is intersected with medication oral factor Xa before being unioned with the VTE prophylaxis received.

Finally, occasionally there is a need to define a local definition or a main foreign expression that can only be used within the query in which it's defined. CQL queries can include a (inaudible) keyword for this purpose.

In this example taken from CMS53, Primary PCI Received within 90 Minutes of Hospital Arrival," the first PTI definition is introduced to establish first PCI procedure that starts on or after the hospital arriving time for the encounter. This allows that named first PCI to be used throughout the rest of the query to refer to that procedure.

In the "Where" clause, we see the first PCI for this encounter must start at 1,440 minutes or less on or after hospital arrival time. There must not be a (inaudible) therapy that starts after the hospital arrival time and before the start of the first PCI.

That's a kind of whirlwind tour of the types of expressions that you'll see and all the different constructs that are used in CQL throughout the EH and CAH measures. At this time, I will turn it back over to Shanna to review the available tools and resources.

Shanna Hartman: Thanks, Bryn.

We wanted to give you guys some available resources if you have additional information about CQL. The CQL Specifications can be found at the HL7 links provided here. This defines the Clinical Quality Language grammar, semantics, and associated specifications to enable the representation and sharing of clinically-focused logic:
The CQL-based HQMF Implementation Guide...this defines the approach to using CQL, the QDM, and HQMF to define eCQMs.
The link to the eCQI Resource Center, ecqi.healthit.gov...these are one-stop shops for the most current resources to support electronic clinical quality improvement. Here you can find current versions of the CDMs and CQL specifications, additional CQL tools, as well as this is where our eCQM specifications list.
The Value Set Authority Center at the link provided here.

The CQL Formatting and Usage Wiki...this is a collaborative workspace for the development of CQL formatting conventions and usage patterns for the representation and logic within quality measures.

The CQL GitHub Tools Repository, which provides tooling support for the standards, including a CQL-ELM translator.

The Measure Authoring Tool that allows measure developers to author eCQMs

The Bonnie Testing Tool, which is designed to support streamlined and efficient pretesting of eCQMs.

And if you have any CQL questions or issues, you can visit our ONC JIRA Issues Tracker; and our CQL project is listed here.

Tricia Elliott: Great, thank you so much, Shanna and Bryn.

We will now open up our Q&A segment. Brenda Kirk asks: "Is 'union' the same as 'or'?"

Bryn, can I defer that to you?

Bryn Rhodes: Absolutely.

"Union" is conceptually the same as "or," but it serves a different purpose within CQL. A union works on lists, to combine the lists...the lists of the integers 1, 2, 3 and the list of the integers 4, 5, 6 and can union them together. It's a union operation.

"Or" works on "Yes" or "No." So if I have an expression that returns true false, I can say, "Is this true or is that true?" So they're different types that they operate on. Does that help?

Yes, I believe so. That was great, thank you. There's also a question...a little bit further explanation on what "Return" means.

A query is an expression that returns a result; typically, that's a list of things. So if I say, "My query is just encounters where length in days of the relevant period is 120," then I'm going to get a list of encounters that match that criteria. By default, the result is the set of elements that match the criteria and come from the source.

I can also introduce a Return, and you'll see that used to change what those elements look like in the results. You can say, "Return" instead of the encounter, you could return the length in days of the encounter; and what you would get is a list of the links of the encounters rather than a list of the encounters themselves.

Tricia Elliott: We do have a question from Shane Fenton at Adventist regarding: "Our healthcare facility is using or planning to use this language to standardize internal metrics." Do you have any perspective on that, Bryn or Shanna?

Bryn Rhodes: The language and specifications are certainly built so that they could; whether they do is, obviously, an internal decision. But that's certainly something that the specifications support, and there's tooling and even products that are coming online that will help support hospitals doing exactly that.

Okay, we have a question about the Coalesce function; so, Bryn, this is probably back to you: How the expressions are ordered...is there any logic to it? In the example, Relevant Period comes before Author Date/Time, and the other functions related to Coalesce.

The Coalesce takes the expressions in the order they're retrieved or in the order they're listed in the expression. So in that case, we'd put Relevant Period first because if there's a relevant period, we'd want to use that. Only in the case that there's *not* a relevant period do we want to pick up the Author Date/Time.

Tricia Elliott: We have a question requesting confirmation that 2018 data submitted in 2019 will not have to use CQL, and that is correct. The 2018 data that is due to CMS in February and to the Joint Commission in March will still be in the QDM model. It's with discharges on or after January 1st that it will be CQL logic.

We have a question coming in: "What is the best reference for those interested in writing eCQMs? Is it the CQL User Guide? Are there any types of developer trainings?

Shanna, can I give that to you?

Shanna Hartman: Monthly we do host what we call "Cooking with CQL," that is an interactive session that provides how to use CQL. So that is one event that we do host monthly, and you can find that information on the eCQI Resource Center. On the eCQI Resource Center, there's also a CQL Style Guide that is there for use for how to use CQL. Also, feel free to reach out to the eCQI resource center e-mail address; and they can point you in the direction for additional information or helping you get access to those monthly "Cooking with CQL" sessions.

Tricia Elliott: We also post some links on the Joint Commission website as well for the events that we host or we collaborate on with CMS.

Did you have a comment, Bryn?

Bryn Rhodes: Yeah, just to add to what Shanna said...the CQL Formatting and Usage Wiki that the link is included in the slides, that has all the examples for past "Cooking with CQL" sessions; so that's a great resource to go through. Then there are also several topics on that site that just talk about different issues and training that's available.

Tricia Elliott: I think that's all the time we have for today. We do have a few more questions that came in, and we'll address those through FAQs. Some are very specific to some of the logic, so we'll be sure to get responses to those.

This concludes our webinar for today. Thank you to everyone who took the time to present at today's webinar...a huge thank you to our partners at CMS, and ESAC with Bryn and Shanna presenting today. And thanks to all of you who listened. Have a great day!