Cooking with CQL Q&As

Session 61 - Thursday, February 24, 2022

Functions in CQL

**Q:** Regarding aggregate functions in Clinical Quality Language (CQL) using Fast Healthcare Interoperability Resources® (FHIR®) v4.0.1, what is the difference between the CountOfListOfNulls function versus the LengthOfListOfNulls function?

**A:** The CountOfListOfNulls, an aggregate operator, will ignore nulls consistent with the other aggregate operations available in CQL. The Count operator counts the number of non-null elements in a list. Whereas Length is a list operator that returns how many elements are in a list, including nulls.

**Q:** With regard to aggregate functions in Clinical Quality Language (CQL) using Fast Healthcare Interoperability Resources® (FHIR®) v4.0.1, considering the expression define EmpyList: List<Integer> { }, why does this definition include List<Integer>, where the previous one does not?

**A:** In the define EmptyList: List<Integer> { } expression, `List<Integer>` is the *type specifier* for the list. In a *list selector* in CQL, authors can optionally provide the type of the list (e.g., list of integers 1, 2, 3) in addition to the list itself. If the expression only provided the `{ }` braces, since there are no elements, the list is of type `List<Any>` (i.e. the list is a list of elements of any type). Since the `Sum` operator is defined only for lists of numeric types (e.g. List<Integer>), it is a type error to try to Sum it. By specifying that the expression is actually a list of Integers, the translator allows the list to be passed to the Sum function. If the list has elements, the type of the list will be inferred based on the type of elements in the list. For a complete reference of the aggregate functions available in CQL on the [CQL webpage](https://cql.hl7.org/09-b-cqlreference.html#aggregate-functions) as well as the Aggregate Functions topic in the [CQL Author’s Guide](https://cql.hl7.org/02-authorsguide.html#aggregate-operators).

Queries in CQL

**Q:** Considering Clinical Quality Language (CQL) aggregate functions using Fast Healthcare Interoperability Resources® (FHIR®) v4.0.1 and calculating the number of each kind of observation a patient has, can the query use the GROUP BY clause?

define NumberOfObservationsPerCode:

[Observation] O

return {

code: O.code,

numberOfObservations: Count([Observation] InnerO where InnerO.code ~ O.code)

}

**A:** No, CQL does not define a *group by* clause like SQL has, but grouping queries can still be expressed using the approach illustrated here. Users may submit feedback requesting the addition of a group by clause in CQL as a shorthand for this type of calculation.