Cooking with CQL Qs & As – Session 35

May 23, 2019 – 4:00-5:00 PM ET

Queries in CQL

Q: When setting up relationships between multiple sources of data at the same time, what should you expect to see as the return with multi-source queries without specifying a return statement? For example, in CMS144v7: Consecutive Heart Rates less than 50 Inpatients, what would you expect to see as the return if not using a return statement?
A: Without a return statement, you'll get a couple of results that have elements for each of the sources. So, you'll get heart rate and a moderate or severe LVSDHF (left ventricular systolic dysfunction heart failure) inpatient encounter:

```
// Result Type:
{
  { HeartRate: { id: 'obs-1', code: ... }, ModerateOrSevereLVSDHFInpatientEncounter: { id: 'enc-1', code: ... } }
  { HeartRate: { id: 'obs-2', code: ... }, ModerateOrSevereLVSDHFInpatientEncounter: { id: 'enc-1', code: ... } }
  { HeartRate: { id: 'obs-3', code: ... }, ModerateOrSevereLVSDHFInpatientEncounter: { id: 'enc-2', code: ... } }
  { HeartRate: { id: 'obs-4', code: ... }, ModerateOrSevereLVSDHFInpatientEncounter: { id: 'enc-2', code: ... } }
  ...
}
```

If you don't specify a return, then for each of the items in your "from," you'll get a result. The actual result of the query will be a list where you'll have every combination of heart rate and moderate or severe LVSDHF inpatient encounter for this patient. A return is typically used with a multi-source query to pick out particular elements to be returned. The "where" clause eliminates combinations of those encounters so it would pull out the rows that don't match that criteria. In the end, you only end up with rows that match the criteria, but since you didn't specify the return, you will still get this list. Typically, if the result is used elsewhere, then a return would be included. If you're only using it in something like an "exists", then you don't need to include a return.

Q: In multi-source queries, if an attribute is not mentioned specifically in the query, like the "method" attribute, will it still be included in the return?
A: Yes, if it is defined as an attribute in that source. For example, "method" is one of the attributes for "Physical Exam, Performed". So, for the default behavior (when no return is specified), each element will have the same structure as the source it came from.

Q: In multi-source queries, is it feasible to specify the type of attributes we want returned?
A: Yes, you can use return, it's an arbitrary expression. So whatever you want to return from the "from," you can.

Using Quality Data Model (QDM)

Q: QDMs are not really used for interoperability like FHIR is, it defines the type of metadata that we call attributes. Is it true that EHRs have been implemented to be able to export the full QDM set of attributes every time you ask for something? That's the ideal.
A: In typical implementation, that's the structure seen. A typical scenario is a database with columns that match to attributes in QDM so the overall structure will look like that but not all of the columns will necessarily be filled. From an architectural perspective, this would be a straightforward and reasonable way to do the implementation.

Time Calculations

Q: Referring to CMS52v7, does the laboratory test always have a relevant period?
A: It does. It also has a resultDatetime, the time the result was actually reported. The authorDatetime is used to provide the negation time when representing a laboratory test not performed, but resultDatetime is used to indicate when it became available or posted and the relevantDatetime is used for the physiologic time, when it was drawn. There are some specimens that are obtained over a period of time.

Q: With regard to authorDatetime, if that's the time it's entered into the clinical software then what happens when someone goes back to edit it? Is it the last modified time or the original creation time and do we have a way of handling that?
A: We're not sure as that has not come up before. There's an amended time, which would still be the recorded time, so it may depend on your use case. The relevantDatetime would still be the dispense time. QDM doesn't say how to handle it, but we may want to specify if it was the original time or amended time – we'd likely have to define it.