2018-05-09 - SLPG Meeting

Date & Time
20:00 UTC Wednesday 9th May 2018

Teleconference Details
To join the meeting please go to https://snomed.zoom.us/j/471420169.
Further information can be found at SLPG meeting information

Goals
- Progress SNOMED Query language
- Future goals:
  - Transitive relationships in ECL
  - Ability to execute maps from within ECL
  - URI standard

Attendees
- Chair: Linda Bird
- Project Group: Michael Lawley, Ed Cheetham, Andrew Perry, Anne Randorff Højen, Rob Hausam

Apologies
- Allow nested where, version, language
- Scope of variables is inner query

<table>
<thead>
<tr>
<th>Description</th>
<th>Owner</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome and apologies</td>
<td>Linda Bird</td>
<td></td>
</tr>
<tr>
<td>Query Language - Recap from previous meetings</td>
<td>Linda Bird</td>
<td>Examples: version and language</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- &lt;&lt; 64572001 [Disease] ([{term = &quot;heart&quot;]}) VERSION <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- &lt;&lt; 64572001 [Disease] ([{synonym = &quot;heart&quot;]}) VERSION <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- &lt;&lt; 64572001 [Disease] ([{FSN = &quot;heart&quot;]}) VERSION <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- &lt;&lt; 64572001 [Disease] ([{preferredTerm = &quot;heart&quot;]}) VERSION <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a>, LANGUAGE W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- &lt;&lt; 64572001 [Disease] ([{acceptableTerm = &quot;heart&quot;]}) VERSION <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a>, LANGUAGE Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- (* [{term = &quot;heart&quot;]}) VERSION <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a>, LANGUAGE W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- X MINUS Y WHERE X = <em>, Y = (</em> [{term = &quot;heart&quot;]}) VERSION <a href="http://snomed.info/sct/900000000000207008/version/20170731">http://snomed.info/sct/900000000000207008/version/20170731</a>, LANGUAGE W</td>
</tr>
<tr>
<td>Notes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Allow nested where, version, language</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Scope of variables is inner query</td>
</tr>
</tbody>
</table>
Examples: where

- X MINUS >! X WHERE X = (<< 1234 : 5678 = << 6547)
- X MINUS >! X WHERE X = (<< 1234 : 5678 = << 6547) VERSION http://snomed.info/sct/900000000000207006/version/20180131
- X MINUS >! Y WHERE X = (<< 1234 : 5678 = << 6547), Y = (<< 1456) VERSION http://snomed.info/sct/900000000000207006/version/20180131
- X MINUS >! X WHERE X = (<< 1234 : 5678 = << 6547) VERSION http://snomed.info/sct/900000000000207006/version/20180131
- X MINUS >! X WHERE X = (<< 1234 : 5678 = << 6547), LANGUAGE 900000000000508004 [GB English]
- X MINUS >! X WHERE X = (<< 1234 : 5678 = << 6547) VERSION http://snomed.info/sct/900000000000207006/version/20180131, LANGUAGE 900000000000508004 [GB English]
- X minus >! X WHERE X = ( < M WHERE M = ( < 1234)) VERSION http://snomed.info/sct/900000000000207006/version/20180131, LANGUAGE 9000018810000000108 [GB clinical extension LRS], 900000000000508004 [GB English]

Notes

- Allow nested variable definitions, but recommend that people don’t due to readability
- Scope of variables is the inner query
- No recursion e.g X WHERE X = 1234 MINUS X
  - ie can't use a variable in its own definition
  - ie X is only known on the left of the corresponding WHERE, and not on the right of the WHERE
### Filters for Lexical Searching

**Linda Bird**

What filter keywords will we introduce for Term-based searching, and what are their exact meanings?

- **D.term**
  - D.term = "*heart*"
  - D.term = wild:"*heart*"
  - D.term = regex:"*heart.*"
  - D.term = match:"hear att"
  - D.term = (sv) wild: "*heart*"

- **D.languageCode**
  - D.languageCode = "en"
  - D.languageCode = "es"

- **D.caseSignificanceId**
  - D.caseSignificance = 9000000000000448009 |entire term case insensitive|
  - D.caseSignificance = 900000000000017005 |entire term case sensitive|
  - D.caseSignificance = 900000000000020002 |only initial character case insensitive|

- **D.caseSignificance**
  - D.caseSignificance = "insensitive"
  - D.caseSignificance = "sensitive"
  - D.caseSignificance = "initialCharInsensitive"

- **D.typeId**
  - D.typeId = 900000000000003001 |fully specified name|
  - D.typeId = 900000000000013009 |synonym|
  - D.typeId = 900000000000550004 |definition|

- **D.type**
  - D.type = "FSN"
  - D.type = "synonym" 
  - D.type = "textDefinition"

- **D.acceptabilityId**
  - D.acceptabilityId = 900000000000549004 |acceptable|
  - D.acceptabilityId = 900000000000548007 |preferred|

- **D.acceptability**
  - D.acceptability = "acceptable"
  - D.acceptability = "preferred"

### Additional Syntactic Sugar

- **FSN**
  - FSN = "*heart"
  - D.term = "*heart", D.type = "FSN"
  - FSN = "*heart" LANGUAGE X
  - D.term = "*heart", D.typeId = 900000000000003001 |fully specified name|, acceptabilityId = *

- **synonym**
  - synonym = "*heart"
  - D.term = "*heart", D.type = "synonym"
  - synonym = "*heart" LANGUAGE X
  - D.term = "*heart", D.typeId = 900000000000013009 |synonym|, (D.acceptabilityId = 900000000000549004 |acceptable| OR D.acceptabilityId = 900000000000548007 |preferred|)

- **synonymOrFSN**
  - synonym = "*heart" OR FSN = "*heart"
  - D.term = "*heart", D.type = "synonym" OR D.type = "fullySpecifiedName"

- **textDefinition**
  - textDefinition = "*heart"
  - textDefinition = "*heart", D.typeId = 900000000000550004 |definition|, D.acceptabilityId = *

- **Unacceptable Terms**
  - (D.term = "*heart") MINUS (D.term = "*heart", (D.acceptability = "acceptable" OR D.acceptability = "preferred") LANGUAGE X)
How do we support language preferences, which are defined over multiple language reference sets? For example:

- **Suggestion**: Use `+` to indicate additive (except for PT), and `,` to indicate priority order (with concept-level override). For example:
  - `LANGUAGE 999001891000000105 [Paediatric neurodisability outpatient diagnosis language reference set], 9000000000000508004 [GB English]
  - Priority order: This means that if a concept has descriptions in the Paediatric LRS, then this LRS is used. But if a concept has no descriptions in the Paediatric LRS, then GB English is used
  - `LANGUAGE 999001881000000108 [GB clinical extension LRS], 9000000000000508004 [GB English]
  - Additive: This means that if a concept has a PT in both LRSs, then the PTs in the Paediatric LRS take priority, and the PT in the GB English LRS becomes acceptable. Other terms are acceptable if they are acceptable in either LRS.
  - `LANGUAGE 999001891000000105 [Paediatric neurodisability outpatient diagnosis language reference set],
    999001881000000108 [GB clinical extension LRS], 9000000000000508004 [GB English]
  - Priority order and Additive: This means that if a concept has a PT in both LRSs, then the PTs in the Paediatric LRS take priority, and the PT in the GB English LRS becomes acceptable. Other terms are acceptable if they are acceptable in either LRS.

Other topics for discussion. For example:

- Query language - Can we de-scope relationship filters?
- URI standard - Any updates on the purpose of SNOMED computable language URIs?
- ECL suggestions - Ability to execute maps in ECL
  - The specific use-case here comes initially from Jeremy and relates to being able to work with inactive concepts via the historical association maps. For example, given an ECL expression, e, that identifies a set of concepts to be used for retrieving patient records, you probably also want to retrieve records for sameAs(e) and replacedWith(e)

The next SLPG meeting will be held in 2 weeks at 20:00 UTC on **Wednesday 23rd May**. Topics for next meeting?