

# GELLO using a table done as a package

Another way to access tabular data is to employ a lookup table structure in a package. This example takes various obstetric ultrasound measured parameters and looks up a calculated gestational age.

First let's do the table model:

```
Package FL_GA_Package

class FLToGestationalAgeMap
  FLMeasurement: Real
  AgeinWeeks: Real

class FLToGestationalAgeTableType extends Sequence (FLToGestationalAgeMap)

EndPackage
```

Then the table declaration:

-- Table Declaration

```
Let FLToGestationalAgeTable:
  FLToGestationalAgeTableType =
    Sequence{
      FLToGestationalAgeMap{FLMeasurement = 1.0, AgeinWeeks = 12.8},
      FLToGestationalAgeMap{FLMeasurement = 1.1, AgeinWeeks = 13.1},
      FLToGestationalAgeMap{FLMeasurement = 1.2, AgeinWeeks = 13.4},
      FLToGestationalAgeMap{FLMeasurement = 1.3, AgeinWeeks = 13.6},
      FLToGestationalAgeMap{FLMeasurement = 1.4, AgeinWeeks = 13.9},
      FLToGestationalAgeMap{FLMeasurement = 1.5, AgeinWeeks = 14.2},
      FLToGestationalAgeMap{FLMeasurement = 1.6, AgeinWeeks = 14.5},
      FLToGestationalAgeMap{FLMeasurement = 1.7, AgeinWeeks = 14.8},
      FLToGestationalAgeMap{FLMeasurement = 1.8, AgeinWeeks = 15.1},
      FLToGestationalAgeMap{FLMeasurement = 1.9, AgeinWeeks = 15.4},
      FLToGestationalAgeMap{FLMeasurement = 2.0, AgeinWeeks = 15.7},
      FLToGestationalAgeMap{FLMeasurement = 2.1, AgeinWeeks = 16.0},
      FLToGestationalAgeMap{FLMeasurement = 2.2, AgeinWeeks = 16.3},
      FLToGestationalAgeMap{FLMeasurement = 2.3, AgeinWeeks = 16.6},
      FLToGestationalAgeMap{FLMeasurement = 2.4, AgeinWeeks = 16.9},
      FLToGestationalAgeMap{FLMeasurement = 2.5, AgeinWeeks = 17.2},
      FLToGestationalAgeMap{FLMeasurement = 2.6, AgeinWeeks = 17.6},
      FLToGestationalAgeMap{FLMeasurement = 2.7, AgeinWeeks = 17.9},
      FLToGestationalAgeMap{FLMeasurement = 2.8, AgeinWeeks = 18.2},
      FLToGestationalAgeMap{FLMeasurement = 2.9, AgeinWeeks = 18.6},
      FLToGestationalAgeMap{FLMeasurement = 3.0, AgeinWeeks = 18.9},
      FLToGestationalAgeMap{FLMeasurement = 3.1, AgeinWeeks = 19.2},
      FLToGestationalAgeMap{FLMeasurement = 3.2, AgeinWeeks = 19.6},
      FLToGestationalAgeMap{FLMeasurement = 3.3, AgeinWeeks = 19.9},
      FLToGestationalAgeMap{FLMeasurement = 3.4, AgeinWeeks = 20.3},
      FLToGestationalAgeMap{FLMeasurement = 3.5, AgeinWeeks = 20.7}
    }
```

And finally the code. In this example, the input could have come from an archetype and this section is commented out.

```

imports iso_21090_datatypes

/*
Let femurLength: Real = If template.Third_Trimester_Growth_Scan
    .Findings
    .Fetal_Biometry
    .Femur_Length_FL
    .Measurement
    .oclIsDefined() then
template.Third_Trimester_Growth_Scan
    .Findings
    .Fetal_Biometry
    .Femur_Length_FL
    .Measurement
    .value.oclAsType(PQ)
    .value
    else null
endif

*/

Let femurLength: PQ = PQ{value = 3.2, unit='cm'}

If (femurLength.value >= FLToGestationalAgeTable.first().FLMeasurement) and (femurLength.value <=
FLToGestationalAgeTable.last().FLMeasurement) then

    FLToGestationalAgeTable->select(o | femurLength.value >= o.FLMeasurement).last().AgeinWeeks.toChar()

else
    ''
endif

```

Here is how that looks in Result Explorer:

Results Explorer		
Name	Class	Data
FLToGestationalAgeTable[26]	FLToGestationalAgeTable...	Sequence{<FLToGestationalAgeMap: TRTTIClone_FLToGestationalAgeMap>, <FLT...
femurLength	PQ	3.2 cm
_Result_	String	19.6