Draft 2019-09-30

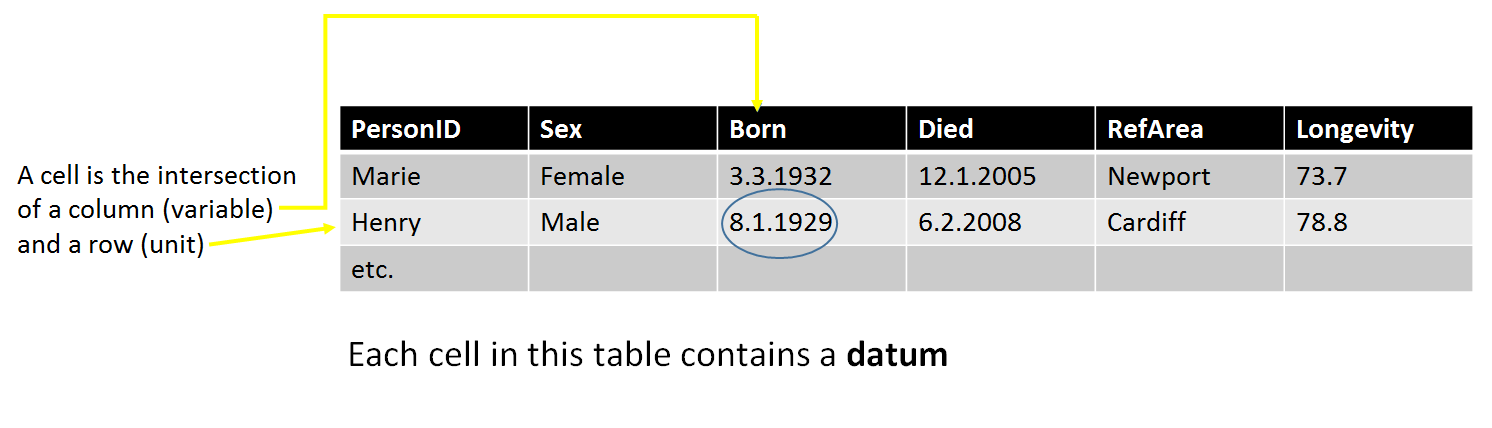
### Unit record data structure (Wide format)

A Unit record data table, as shown in Figure 3, is a common way to organize data. This structure is also referred to as a rectangular data file.

### 

*Figure 3 . Unit record data table*

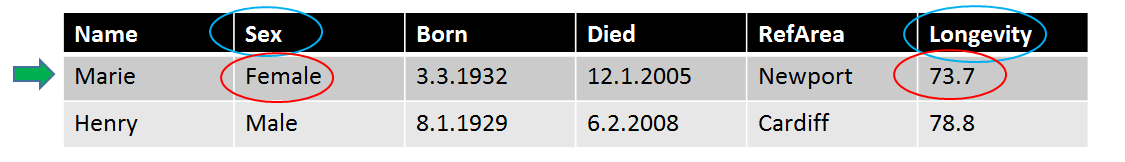
A cell in the Unit record table is an intersection between a column representing a variable and a row representing a measurement unit. Each cell of the table contains a Datum.



*Figure 4. Unit record data cell*

The objects of the Wide format Unit record data table are Unit records, Variables and DataValues.

In the Wide format the rows correspond to each unit record, which is a set of DataValues for one entity. The columns correspond to each variable measure or categorization. Cell entries are DataValues.

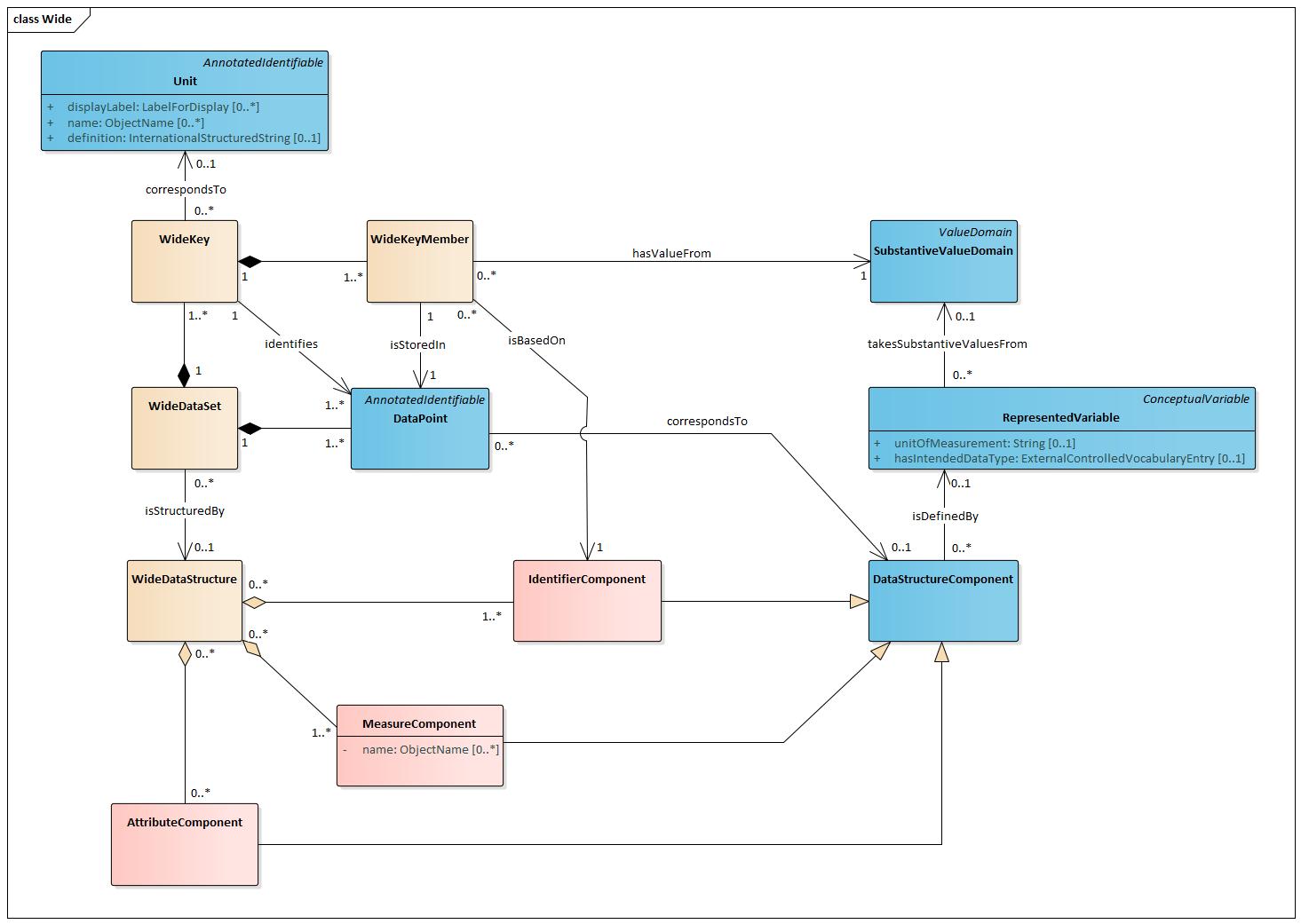


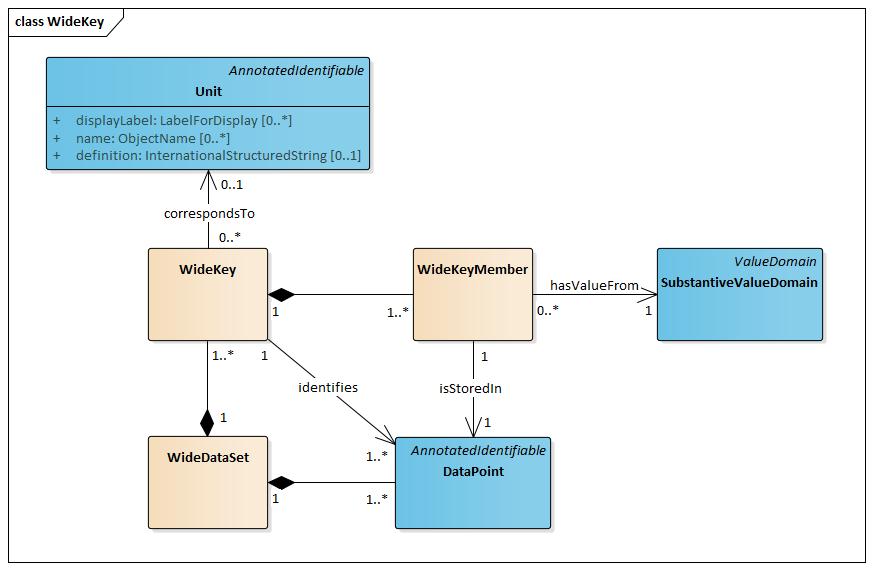
*Figure 5. Wide format objects*

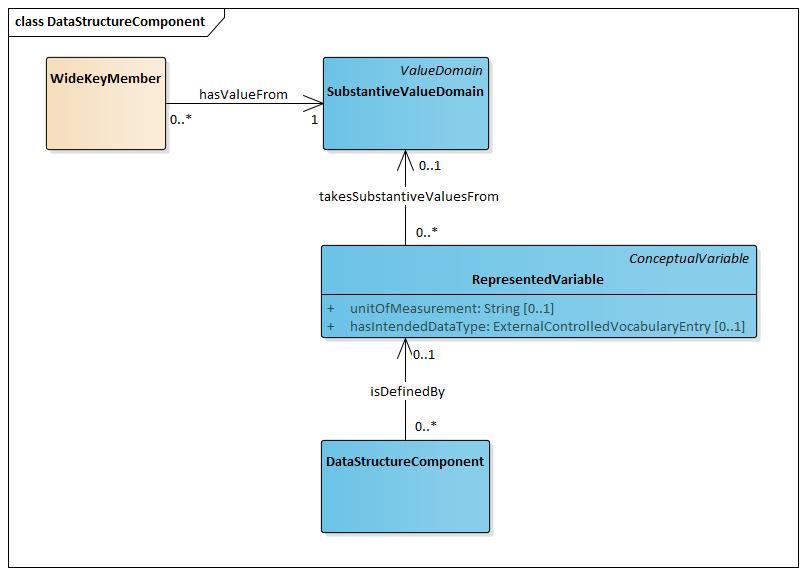
In figure 5 above ‘Marie’ and ‘Henry’ are identifiers for each of the records. ‘Sex’, ‘Longevity’ etc. are variables and ‘Female’ and ’73,7’ are example of DataValues.

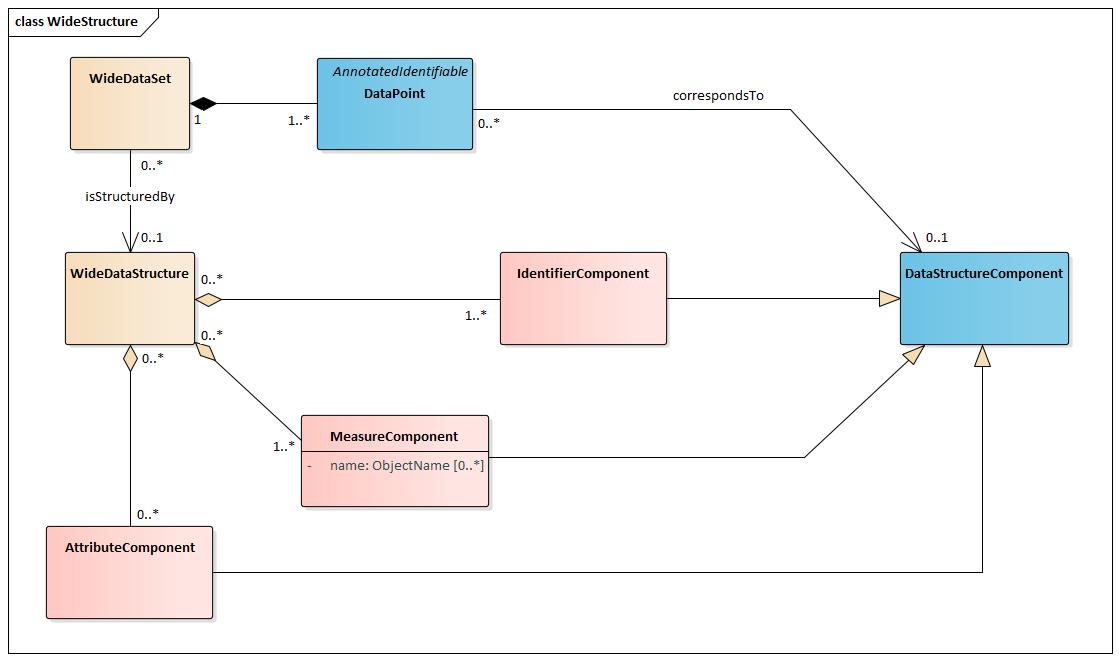
# Unit record data structure (wide format)

The figure below shows the DDI4 classes used to represent unit data in wide format. This is probably the most common layout for data – the traditional table of data as used in many statistical packages and spreadsheet programs. Columns are variables and each row contain the DataPoints for one Unit. The model for the Wide structure is shown below.



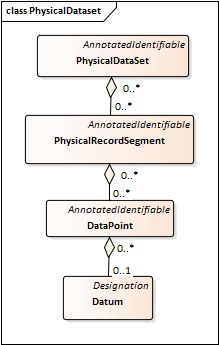
The WideDataSet contains DataPoints, all the “cells” in the table. Some of these contain values for keys that identify the DataPoints common to an individual row of the table. In the example dataset above the “Name” column contains DataPoints that contain the key values that identify a row and also correspond to a Unit. The DataPoint in the upper left of the table contains the key value “Marie”. That DataPoint identifies the other DataPoints also associated with the person named “Marie”, the DataPoints in the first row of the table. A WideKey can be composed of more than one WideKeyMember. Our table might have, for instance, contained another column like “Family” so that we could identify the Marie in a particular family.

A row of the table is also further structured by DataStructureComponents. These are defined by RepresentedVariables, which in turn provide the SubstantiveValueDomain (e.g. a Codelist) for a WideKeyMember.

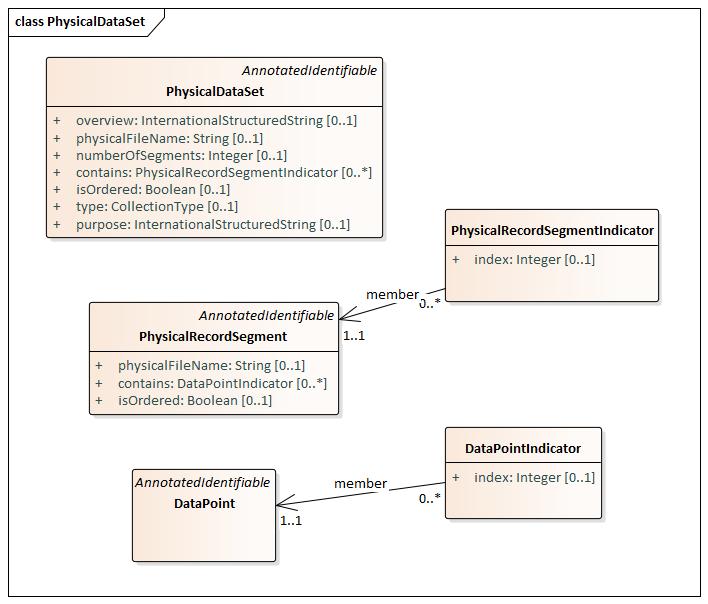
 DataStructureComponents are of three types. An IdentifierComponent is the basis for a WideKeyMember that contributes to identifying which DataPoints belong to the same row. A MeasureComponent identifies the DataPoints within each row that contain a measure of interest. An AttributeComponent provides context for the MesureComponents. This could be, for example, time or condition of data collection.

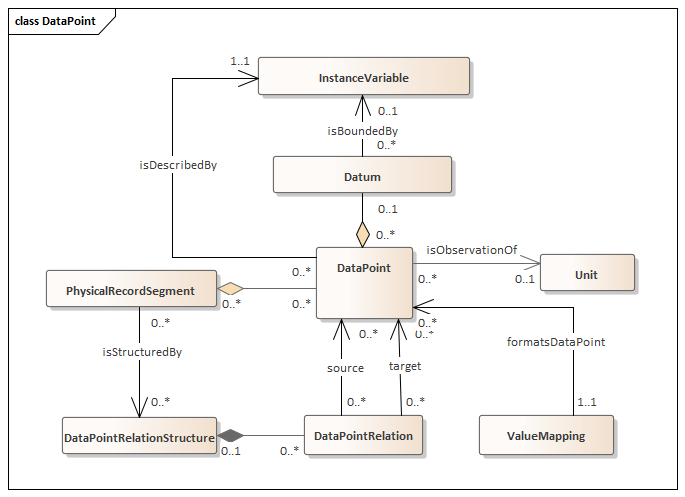
DDI4 still contains the construct of a LogicalRecord. This is a collection of InstanceVariables. These correspond to the RepresentedVariables defining the DataStructureComponents.

DDI4 models a physical representation of the structure, starting from the lowest level up, with PhysicalRecordSegments containing DataPoints. A LogicalRecord may have multiple associated record segments. This was more common historically when there were serious constraints on the length of physical records (like 20 column cards). It is more common now to have just one record segment per LogicalRecord.

A Wide PhysicalDataSet is a collection of DataPoints organized into PhysicalRecordSegments. A PhysicalRecordSegment has a layout described by an association to a PhysicalSegmentLayout. The latter contains information like whether the segment is comma delimited, what line terminator it uses, what escape character it uses, and so on.

The PhysicalDataset



The DataPoint ties a lot of classes together. DataPoints can be thought of as containers for Datums. The DataPoint is also described by an InstanceVariable. A PhysicalRecordSegment contains a set of DataPoints. PhysicalRecordSegments tie the Datum into a larger structure. PhysicalRecordSegments are structured by a DataPointRelationStructure through a set of DataPointRelations. This approach allows complete flexibility in describing the ordering of DataPoints in a PhysicalRecordSegment.

The DataPoint also has an associated ValueMapping. This provides specific information, such as the number of decimal positions, about the physical representation of a Datum for the DataPoint.

Finally, a DataPoint is associated with a Unit. Again, think of a spreadsheet. The column for a cell is associated with an InstanceVariable. The row is associated with some entity (a Unit in the DDI model).

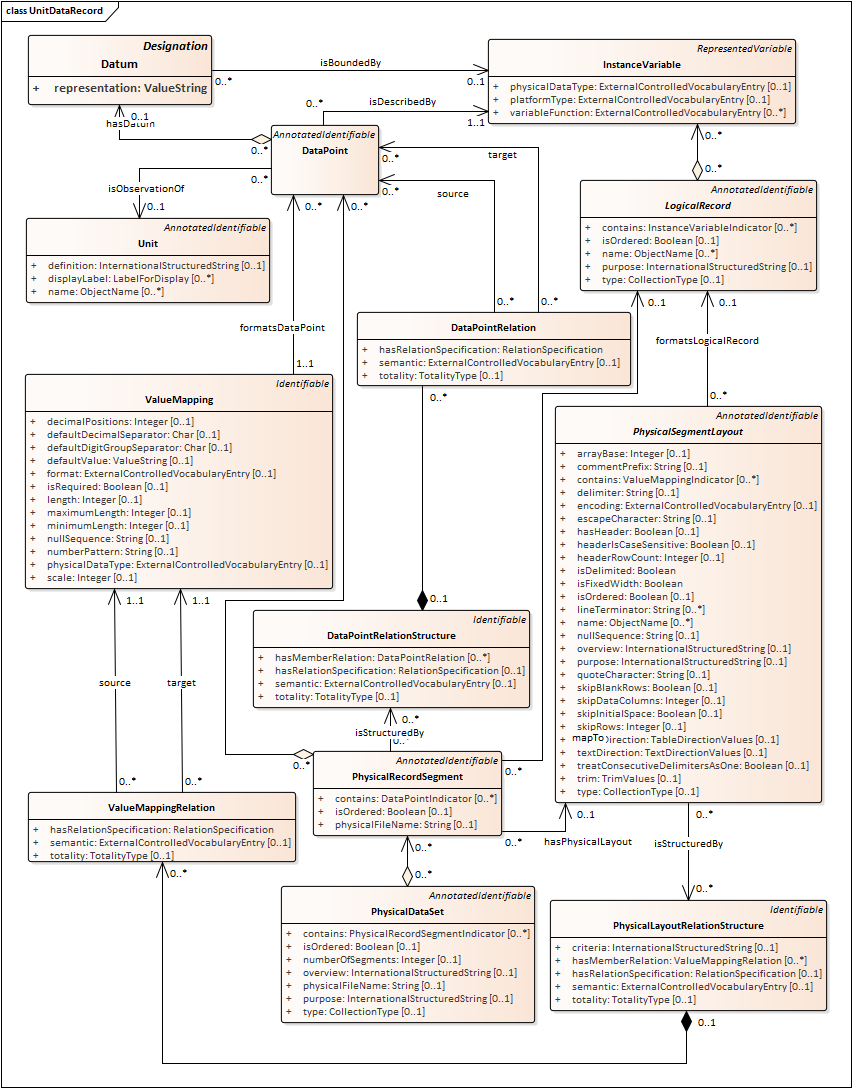


Figure 14 DDI4 model for Unit Data Records showing properties for all classes