

How to use SUPPQUAL for specifying natural key variables in define.xml?

Introduction

- ▶ High quality metadata
 - List of key variables
 - Standard SDTM and SEND variables
 - SUPPQUAL datasets
- ▶ Lack of technical guidance on how to reference keys from SUPPQUAL

Key variables

- ▶ CDISC Define-XML standard does not include implementation guidance like SDTM IG
 - PhUSE “Define-XML Version 2.0 Completion Guidelines”
- ▶ Best knowledge source for key variables are SDTM/SEND IGs:
 - data definition file should describe each dataset by providing dataset’s *natural key* structure
 - “A natural key is a set of data (one or more columns of an entity) that uniquely identifies that entity and distinguishes it from any other row in the table. The advantage of natural keys is that they exist already; one does not need to introduce a new, “unnatural” value to the data schema.” (SDTM IG 3.3 section 3.2.1.1)”

Natural key variables

- ▶ Physical structure of dataset
- ▶ Consistent with description of dataset structure

Tabulation Datasets for Study CDISC01 (SDTM-IG 3.1.2)			
Dataset	Description	Structure	Keys
LB	Laboratory Tests Results	One record per analyte per visit per subject	STUDYID, USUBJID, LBCAT, LBMETHOD, LBTESTCD, LBDTC, VISITNUM

- key variables specify “*analyte*” as a combination of values in LBCAT, LBMETHOD and LBTESTCD variables

- ▶ Sorting order of records

Surrogate key variables

- ▶ Single-part, artificially established identifier for a record
 - For example, --SEQ
- ▶ Great for linking records across datasets
 - SUPPQUAL
 - RELREC
- ▶ Should not be used as key variables!
 - There are exclusions like TS domain

SUPPQUAL datasets

- ▶ Special method to utilize non-standard variables
 - QNAM – variable name
 - QLABEL – variable label
 - Linkage to parent domain by RDOMAIN, USUBJID, IDVAR and IDVARVAL variables
 - For example, IDVAR='AESEQ'

Variable metadata in define.xml

- ▶ *ItemRef* element
 - Variable info in relation to a dataset
 - Order number, mandatory, etc.
 - Provided inside *ItemGroupDef* element which describes a dataset
- ▶ *ItemDef* element
 - Additional info about variable
 - Name, label, origin, codelist, etc.
- ▶ Key variables is dataset-level info
 - Provided in *ItemRef* element by *KeySequence* attribute

Example of STUDYID, USUBJID keys in DM domain

```
<!-- Dataset Definition (VS) -->
<ItemGroupDef OID="IG.DM"
  Domain="DM"
  Name="DM"
  Repeating="No"
  IsReferenceData="No"
  SASDatasetName="DM"
  Purpose="Tabulation"
  def:Structure="One record per subject"
  def:Class="SPECIAL PURPOSE"
  def:ArchiveLocationID="LF.DM">
  <Description>
    <TranslatedText xml:lang="en">Demographics</TranslatedText>
  </Description>
  <ItemRef ItemOID="IT.DM.STUDYID" OrderNumber="1" Mandatory="Yes"
    KeySequence="1"/>
  <ItemRef ItemOID="IT.DM.DOMAIN" OrderNumber="2" Mandatory="Yes"/>
  <ItemRef ItemOID="IT.DM.USUBJID" OrderNumber="3" Mandatory="Yes"
    KeySequence="2" MethodOID="MT.DM.USUBJID"/>
  <ItemRef ItemOID="IT.DM.SUBJID" OrderNumber="4" Mandatory="Yes"/>
  <ItemRef ItemOID="IT.DM.RFSTDTC" OrderNumber="5" Mandatory="No"
    MethodOID="MT.DM.RFSTDTC"/>
  <ItemRef ItemOID="IT.DM.RFENDTC" OrderNumber="6" Mandatory="No"
    MethodOID="MT.DM.RFENDTC"/>
```

Non-standard variables from SUPPQUAL

- ▶ Specified using Value Level Metadata (VLM) in define.xml
- ▶ Similar to regular variables (*ItemGroupDef*), new non-standards variables from SUPPQUAL are listed in VL list (*def:ValueListDef* element)
- ▶ To provide a reference to key in SUPPQUAL, assign *KeySequence* attribute to specific VL item in VL list

Example of non-standard key LBCLSIG variable for LB domain

```
<def:ValueListDef OID="VL.SUPPLB.QVAL">
  <ItemRef ItemOID="IT.SUPPLB.QVAL.LBCLSIG"
    OrderNumber="1"
    Mandatory="No"
    MethodOID="MT.CLSIG"
    KeySequence="8">
    <def:WhereClauseRef WhereClauseOID="WC.SIPPLB.QNAM.LBCLSIG"/>
  </ItemRef>
```

Challenges

1. Most Define.xml tools do not support keys from SUPPQUAL datasets
2. Outdated stylesheets do not show keys from SUPPQUAL datasets
3. New stylesheet from PhUSE utilizes specific attribute *Name* as display

Issue #1

- ▶ Many existing define.xml tools do not support keys from SUPPQUAL
 - For example, Pinnacle 21 Community 3.0
- ▶ Solution: edit define.xml manually
 - XML Spy, Notepad+ (recommended option), or Notepad

Steps

1. Finalize define.xml
2. Open with editor
3. Search for your SUPPQUAL variable in associated value list
 - Option 1: look for your value list (search for “<def:ValueListRef”)
 - Option 2: search for name of non-standard variable in *ItemOID* value
 - For example, searching for “*REPNUM*” text, you may find something like:

```
<ITEMREF ITEMROID="IT.SUPPVS.QVAL.REPNUM" ...
```

OR

```
<ITEMREF ITEMROID="IT.SUPPVS.QVAL.SUPPVS.QNAM.EQ.VSREPNUM-1091FC31"...
```

1. Add *KeySequence*="NNN" to *ItemRef* element, where *NNN* is the key sequence number
2. Save define.xml and open in browser to test

Example of adding VSREPNUM variable as key

```
<ItemRef ItemOID="IT.SUPPVS.QVAL.SUPPVS.QNAM.EQ.VSREPNUM-1091fc31"  
  OrderNumber="1" Mandatory="No" MethodOID="MT.SUPPVS.QNAM.VSREPNUM"  
  KeySequence="6">  
  <def:WhereClauseRef WhereClauseOID="WC.SUPPAE.QNAM.EQ.TRTEMFL"/>  
</ItemRef>
```

Dataset	Description	Class	Structure	Keys	Location
VS	Vital Signs	FINDINGS	One record per vital sign measurement per time point per visit per subject	STUDYID, USUBJID, VSTESTCD, VISITNUM, VSTPTNUM, QNAM.VSREPNUM	vs.xpt

Issue #2

- ▶ Older stylesheets for define.xml do not support keys from SUPPQUALs

Dataset	Description	Class	Structure	Keys	Location
VS	Vital Signs	FINDINGS	One record per vital sign measurement per time point per visit per subject	STUDYID, USUBJID, VSTESTCD, VISITNUM, VSTPTNUM	vs.xpt

- ▶ Solution: use a new stylesheet published by PhUSE

New stylesheet from PhUSE

- ▶ Published on PhUSE wiki in 2018
- ▶ Many nice features
 - including support for keys from SUPPQUAL
- ▶ Included in P21 Community 3.0.x
 - Previous P21 Community 2.2.0 utilized original (old) stylesheet from CDISC
- ▶ Name of new stylesheet is different
 - Ensure to either
 1. Rename stylesheet file if needed
 2. Edit define.xml file

Editing a name of stylesheet in define.xml

- ▶ Define.xml exported from P21 Community 2.2.0:

```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/xsl" href="define2-0-0.xsl"?>
```

- ▶ A file name of new stylesheet from PhUSE is “*define2-0.xsl*”

```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/xsl" href="define2-0.xsl"?>
```

Issue #3

- ▶ Expected view of a reference to key from SUPPQUAL
 - “*QNAM.VSREPNUM*”
- ▶ Actual
 - “*QNAM.QVAL.SUPPVS.QNAM.EQ.VSREPNUM-1091fc31*”
 - or
 - “*SUPPVS.QNAM.EQ.ce8eb9c250eaa8deec1334a597a941cea6741b51*”

Dataset	Description	Class	Structure	Keys	Location
VS	Vital Signs	FINDINGS	One record per vital sign measurement per time point per visit per subject	STUDYID, USUBJID, VSTESTCD, VISITNUM, VSTPTNUM, QNAM. SUPPVS.QNAM.EQ.ce8eb9c250eaa8deec1334a597a941cea6741b51	vs.xpt

Why?

- ▶ Stylesheet generates a reference based on attribute *Name* in element *ItemDef*
- ▶ Name value for variable is simple concept
- ▶ However, it is challenging auto-generate meaningful Name value for Value Level Items
 - It may be simple only for SUPPQUAL – use QNAM
 - Example 1: VLM in LB domain with condition based on LBTESTCD
 - Is LBTESTCD a good candidate for Name of VL items?
 - What about a case, when VLM is not limited to a single variable in LB?
 - The same condition may be applied to multiple variables (LBORRES, LBORRESU, LBSTRESN, LBSTRESU)
 - Using LBTESTCD values for Name attribute will produce duplicate names for different VL items

Why? (cont.)

- Example 2: multiple conditions in WhereClause with reference to different variables (e.g., LBTESTCD, LBCAT, LBMETHOD or VSTESTCD, DM.COUNTRY)
- Example 3: other than “EQUAL” condition (e.g., “IN”, “NOT IN”, etc.).
- ▶ Historically, most define.xml tools used some algorithms to generate unique Name attribute for each VL item
 - Random: “*SUPPVS.QNAM.EQ.ce8eb9c250eaa8deec13a597a941cea6741b51*”
 - More short or meaningful: “*SUPPVS.QNAM.EQ.VSREPNUM*”
- ▶ Stylesheet takes these values and attaches them to prefix “*QNAM.*” to generate a reference

Edit *Name* attribute manually

- ▶ Until existing algorithms in define.xml are adjusted

```
<ItemDef OID="IT.SUPPVS.QVAL.SUPPVS.QNAM.EQ.VSREPNUM-1091fc31"  
  Name="QVAL.SUPPVS.QNAM.EQ.VSREPNUM-1091fc31"  
  DataType="integer"  
  Length="2">  
<def:Origin Type="Derived" />  
</ItemDef>
```

```
<ItemDef OID="IT.SUPPVS.QVAL.SUPPVS.QNAM.EQ.VSREPNUM-1091fc31"  
  Name="VSREPNUM"  
  DataType="integer"  
  Length="2">  
<def:Origin Type="Derived" />  
</ItemDef>
```

When to use keys from SUPPQUAL

- ▶ Outdated version of SDTM
 - SDTM 1.7 (IG 3.3) has 30 new variables
 - --REPNUM (Repetition Numbers) is a common key in VS domain
 - in older versions of SDTM, --REPNUM can be stored in SUPPQUAL
 - Historical examples
 - --LAT and --DIR variables introduced in SDTM-IG 3.1.3

When to use keys from SUPPQUAL (cont.)

- ▶ CDISC TAUG with provisional variables
 - 30+ existing TAUGs
 - Provisional variables to be promoted to standard variables in new versions
- ▶ Study-specific non-standard variables
 - CDISC STDM is expanding (e.g., Finding 63 → 91 std. variables)
 - There is always a need for new variables
 - SDTM --LAT (Laterality) within a body like “*Left Leg*”, “*Right Leg*”
 - In pre-clinical studies: additional concept of Laterality within an organ like “*Left side of Liver*”

When to use keys from SUPPQUAL (cont.)

- ▶ 200-char limitation of SAS® XPORT format
 - Example
 - DVTERM (Protocol Deviation Term) is collected as free-text and may be really long
 - Collected value is split across standard DVTERM variable and additional vars in SUPPDV
 - Formally, Protocol Deviation Term info should be represented by all its component variables like “*DVTERM, QNAM.DVTERM1, QNAM.DVTERM2, QNAM.DVTERM3*”

When to NOT use keys from SUPPQUAL

- ▶ SDTM Model vs. practical use of study data
 - Some SDTM variables allows flexibility in their usage and may be utilized instead of SUPPQUALs
 - --SPID (Sponsor-Defined Identifier)
 - SDTM IG acknowledges that -SPID may also have meaning across domains
 - Historically, many users utilized VSSPID to track repetitive measurements for VS assessments instead of using SUPPQUAL
 - Keeping important information in parent domain instead of SUPPQUAL
 - Formal violation of SDTM Model
 - benefits for keeping all core information together drives users toward this implementation

Current tools support for SUPPQUALs

- ▶ Limited
 - Not aware of any tools which utilize keys from SUPPQUALs
- ▶ Humans are still major consumers of this info
 - Good stylesheet with correct display of study metadata in human-friendly way

Summary & Questions

- ▶ The industry's increasing experience with SDTM/SEND implementation drives users to handle new challenges
- ▶ New stylesheet from PhUSE support visual representation of keys from SUPPQUAL
- ▶ While define.xml tools are still catching up with new functionality, users can easily edit their define.xml file to utilize keys from SUPPQUAL
- ▶ While correct implementation of SDTM model is important, users should be aware that today support for keys from SUPPQUALs is still limited

Name: Sergiy Sirichenko
Organization: Pinnacle 21
Address: 1777 Sentry Parkway W, Bldg 17, Ste 405
City, State ZIP: Blue Bell, PA 19422
Work Phone: +1.570.817.6137
E-mail: sergiy@pinnacle21.com
Web: pinnacle21.com