PharmaSUG2011 - Paper PO09

Method to Derive Reproducible SDTM Relationship Datasets

Suwen Li, Everest Research Services Inc., Markham, ON.
Sai Ma, Everest Research Services Inc., Markham, ON.
Regan Li, Everest Research Services Inc., Markham, ON.
Bob Lan, Everest Research Services Inc., Markham, ON.

ABSTRACT

SDTM has relationship datasets SUPPQUAL, CO, and RELREC that are related to parent domains and records. When those relationship datasets are derived from raw data, attention should be used to ensure that the records in those datasets can be traced back to split parent domains correctly. This paper describes how to derive the relationship datasets SUPP--, CO, and RELREC, and employs examples to illustrate the process.

INTRODUCTION

SDTM Implementation Guide [1] described the relationships between records in the relationship datasets RELREC, SUPPQUAL, and CO:

1) Records of one (or more) dataset(s) that are related to record(s) in another dataset (or datasets). Those relationships are represented in RELREC.
2) Non-standard variables that can be related back to parent records in general-observation-class datasets and demographics. Those non-standard variables and their association are captured in SUPPQUAL datasets
3) Relationship between comments in the Comments domain (CO) and parent record(s) in other datasets.

Besides the standard key identifiers STUDYID and USUBJID, the two identifying variables IDVAR and IDVARVAL and domain identifiers RDOMAIN are used in all the three classes of datasets. IDVAR and IDVARVAL are the keys to link records back to parent domains. Although IDVAR can be from many different variables (--SEQ, --GRPID, --REFID, --SPID), it should generally be --SEQ so that the related children records can join back to parent records correctly.

It can be very difficult to derive those SDTM relationship datasets with correct relationship between children records and parent records if SDTM datasets are derived from raw datasets directly. A much easier alternative way can be done by using the idea of SDTM plus [2] as the intermediate parent level datasets. The data structures in SDTM plus is the same as SDTM data, but have some extra variables.

Before deriving the datasets RELREC, SUPPQUAL, and CO, we derive SDTM plus datasets first, which have the required SDTM variables and structures. All other non-standard variables needed in the SUPPQAL datasets or additional variables to identify relationship are also kept in these parent level domains SDTM plus datasets. At the end of each program for an individual domain, if there is a need to generate SUPPQUAL datasets for this domain, the SDTM plus dataset can be split in to a standard SDTM dataset and the corresponding SUPP-- datasets using macro %MSUPP. The macros %MRELREC and %MCO will look for all datasets with variable RELID or COVAL and create RELREC and CO datasets, respectively.

RELREC DATASET

RELREC domain has variables STUDYID, RDOMAIN, USUBJID, IDVAR, IDVARVAL, RELTYPE, and RELID.
The required variable RELID represents the relationship identifier to identify the related records. RELTYPE identifies the hierarchical level of the records in the relationship. It should have value either ONE or MANY. Section 6.3.10.5 reference [1] uses four examples to illustrate how to document the relationships between two datasets using RELREC domain. Although different methods can be used for the same data, the easiest one to implement is the simplest One-to-One combination so that we can always use --SEQ as IDVAR in RELREC domain. Here we use PP and PC domains to describe how to derive the RELREC dataset. RELID was created by USUBJID, PCTEST, and PCDY if necessary or other group categories while we built PC SDTM plus data. RELID is a combination of two related domains PC and PP using a sequence number. The sequence number is the same within the same PCTEST and PCDY or other group categories and will change in different group categories. The PK parameters were

calculated using a non-SAS software and the results were saved in a SAS dataset. The RELID was kept in the resulting SAS dataset and the PP SDTM plus was derived from this SAS dataset. Therefore, RELID was built in both PC and PP SDTM plus datasets. This method ensured the relationship between PC and PP was created correctly. We then used macro %MRELREC to find all datasets containing the variable RELID and generated dataset RELREC.

SDTM plus PC

DOMAIN	USUBJID	PCSEQ	PCTEST	PCORRES	PCORRESU	VISITNUM	VISIT	PCTPT	RELID
PC	002	55	MOXIF	0	NG/ML	1	DAY 9	0	PCPP1
PC	002	56	MOXIF	345.87	NG/ML	1	DAY 9	2	PCPP1
PC	002	57	MOXIF	705.61	NG/ML	1	DAY 9	3	PCPP1
PC	002	58	MOXIF	1173.74	NG/ML	1	DAY 9	4	PCPP1
PC	002	59	MOXIF	1206.78	NG/ML	1	DAY 9	5	PCPP1
PC	002	60	MOXIF	1078.88	NG/ML	1	DAY 9	7	PCPP1
PC	002	61	MOXIF	629.25	NG/ML	1	DAY 9	12	PCPP1
PC	002	63	ESCT	7.3	NG/ML	1	DAY 9	0	PCPP2

SDTM plus PP

DOMAIN	USUBJID	PPSEQ	PPTESTCD	PPORRES	PPORRESU	VISITNUM	RELID
PP	002	1	AUCINF	18502.74	ng*h/mL	1	PCPP1
PP	002	3	AUCT	14872.03	ng*h/mL	1	PCPP1
PP	002	14	CMAX	1206.78	ng/mL	1	PCPP1
PP	002	31	THALF	9.42	h	1	PCPP1
PP	002	35	TMAX	5	h	1	PCPP1
pp	002	36	AUCTAU	219.2	ng*h/mL	1	PCPP2

SDTM RELREC

RDOMAIN	USUBJID	IDVAR	IDVARVAL	RELID
PC	002	PCSEQ	55	PCPP1
PC	002	PCSEQ	56	PCPP1
PC	002	PCSEQ	57	PCPP1
PC	002	PCSEQ	58	PCPP1
PC	002	PCSEQ	59	PCPP1
PC	002	PCSEQ	60	PCPP1
PC	002	PCSEQ	61	PCPP1
PC	002	PCSEQ	62	PCPP1
PC	002	PCSEQ	63	PCPP2
PP	002	PPSEQ	1	PCPP1
PP	002	PPSEQ	3	PCPP1
PP	002	PPSEQ	14	PCPP1
PP	002	PPSEQ	31	PCPP1
PP	002	PPSEQ	35	PCPP1
PP	002	PPSEQ	36	PCPP2

SUPPQUAL DATASETS

SUPPQUAL includes non-standard variables that sponsor collected but not defined in standard SDTM domain datasets, and their association to parent records. Besides the three key variables STUDYID, RDOMAIN, and USUBJID, SUPPQUAL also has variables: QNAM (the name of the Qualifier variable), QLABEL (the label for the variable), QVAL (the actual value for each record), QORIG (the origin of the record), and QEVAL (the evaluator). The QORIG may have value CRF, DERIVED, etc.; the QEVAL may have value SPONSOR, INVESTIGATOR, etc. In the below Vital Signs example, the raw data not only collected standard vital signs test values but also clinical significance results. In the SDTM plus dataset, when we derived standard SDTM variables, the additional variable PCS was remained in VS SDTM plus dataset with variable label 'Clinically Significant'.

SDTM plus VS:

DOMAIN	USUBJID	VSSEQ	VSTESTCD	VSTEST	VSORRES	PCS
VS	002	1	TEMP	Temperature	96.3	NCS
VS	002	2	SYSBP	Systolic Blood Pressure	115	NCS
VS	002	3	DIABP	Diastolic Blood Pressure	65	NCS
VS	002	4	PULSE	Pulse Rate	84	NCS
VS	002	5	RESP	Respiration Rate	18	NCS

We then called the macro %*msupp*(domain=vs, idvar=vsseq, CRF=PCS) to derive SUPPVS dataset.

SDTM SUPPVS:

DOMAIN	USUBJID	IDVAR	IDVARVAL	QNAM	QLABEL	QVALUE	QORIG
VS	002	VSSEQ	1	PCS	Clinically Significant	NCS	CRF
VS	002	VSSEQ	2	PCS	Clinically Significant	NCS	CRF
VS	002	VSSEQ	3	PCS	Clinically Significant	NCS	CRF
VS	002	VSSEQ	4	PCS	Clinically Significant	NCS	CRF
VS	002	VSSEQ	5	PCS	Clinically Significant	NCS	CRF

CO DATASETS

CO dataset captures all the unstructured free text comments. These comments may be related to a subject in general, a specific domain, or a specific record(s) in a domain for a subject. This report only discusses how to derive comments in the last situation that are generally contained in relevant raw data. In the stage of database design, in order to store comments for a specific record(s) in data, more than one variable for comments is always created in case the comments have more than 200 characters. The comments variables in the final raw datasets may have some values or may be all missing if there are no any comments for that data panel or all comments have free texts less than 200 characters. When the SDTM general observation class datasets are derived in SDTM plus, we simply need to rename the comment variables to COVAL, COVAL1, etc. according to the name convention specified in SDTM implementation guide [1]. We developed a macro %CO to create SDTM CO dataset. This macro will find out all domains having COVAL variables and drop COVAL variables that have no any values.

In the below example, that subject has comments on blood sample deviation. When the SDTM plus PC data was derived, the comments were populated in the variable COVAL and COVAL1. The macro %CO was called to find out all datasets having COVAL and derive CO domain from SDTM plus datasets. Because COVAL1 in SDTM plus PC has no any values, it was dropped automatically in the macro.

SDTM plus PC:

USUBJID	DOMAIN	PCSEQ	PCSPID	PCTESTCD	PCORRES	PCDTC	COVAL	COVAL1
002	PC	46	BLSAMP	BDEV	No Deviation	2000-04-29T09:34		
002	PC	47	BLSAMP	BDEV	No Deviation	2000-04-29T09:35		

002	PC	48	BLSAMP	BDEV	Other	2000-04-29T09:36	COMMENTS 1	
002	PC	49	BLSAMP	BDEV	Other	2000-04-29T09:37	COMMENTS 2	
002	PC	50	BLSAMP	BDEV	Other	2000-04-29T09:38	COMMENTS 3	

SDTM CO:

USUBJID	RDOMAIN	IDVARVAL	IDVAR	DOMAIN	COREF	CODTC	COSEQ	COVAL
002	PC	48	PCSEQ	CO	BLSAMP	2000-04-29T09:36	1	COMMENT 1
002	PC	49	PCSEQ	CO	BLSAMP	2000-04-29T09:37	2	COMMENT 2
002	PC	50	PCSEQ	CO	BLSAMP	2000-04-29T09:38	3	COMMENT 3

CONLUSION

The advantage of SDTM plus is that while SDTM general observation classes are derived, all non-standard variables, relationship identifiers, and relationship are presented in the parent domains as well. We can then easily derive the relationship datasets and join them back to corresponding parent records correctly. Another benefit of this method is how it is easy to do validation with it. As all the relationship identifiers and related variables are created in the parent SDTM plus domains, most of the time we only need to validate the SDTM plus.

APPENDIX

MACRO %MRELREC

```
%macro mrelrec;
ods listing close;
ods output Variables=var;
 proc contents data= p._all_;
run;
ods listing;

proc sql noprint;
  /*obtain the number of dataset containing RELID*/
   select count(distinct member) into: tot
   from var
   where index(variable, 'RELID');
quit;

%if &tot>0 %then %do;
proc sql;
   /*assign those dataset name to a series of macro variables*/
   select distinct scan(member, 2,'.') into : mem1-:mem%left(&tot)
   from var
   where index(variable, 'RELID');
quit;

%do i=1 %to &tot;
  /*only one to one relationship, so only --SEQ variable and value kept*/
   %let seq=%left(&&mem&i)seq;
   data _rr&i;
      length idvar $8 rdomain $2;
      set  p.&&mem&i;
      idvar="%left(&&mem&i)SEQ";
      idvarval=strip(put(&seq,best.));
      rdomain=strip("&&mem&i");
      where relid ^=' ';
    keep studyid USUBJID rdomain idvar idvarval relid;
  run;

  %if &i=1 %then %do;
```

```
     data  rr;
        set _rr&i;
     run;
  %end;

  %else %do;
     data  rr;
       set rr _rr&i;
     run;
  %end;
%end;

  data relrec(label="Related Records");
  attrib
  STUDYID LABEL="Study Identifier"    LENGTH=$20
  RDOMAIN LABEL="Related Domain Abbreviation"   LENGTH=$2
  USUBJID LABEL="Unique Subject Identifier"      LENGTH=$200
  IDVAR LABEL="Identifying Variable "     LENGTH=$200
  IDVARVAL LABEL="Identifying Variable Value"      LENGTH=$200
  RELID LABEL="Relationship Identifier"      LENGTH=$20
   ;
   set RR ;
run;

%end;
%mend;
```

MACRO %MSUPP

```
%macro msupp(domain=,
             idvar=,
            CRF=,/*variables from CRF*/
            sponsor=/*variables derived*/ );

%let domain=%upcase(&domain);
%let crf=%upcase(&crf);
%let sponsor=%upcase(&sponsor);

 PROC SORT DATA= p.&domain out=_&domain;
   BY STUDYID USUBJID &idvar;
run;

%if %length(&crf)^=0 %then %do;
data _null_;
   a=tranwrd("&crf", ' ','" "');
  call symput('ncrf',a);
run;
 %let ncrf=&ncrf;
 %let ncrf="&ncrf";
%end;

%if %length(&sponsor)^=0 %then %do;
 data _null_;
   b=tranwrd("&sponsor", ' ','" "');
  call symput('ns',b);
run;
%let ns=&ns;
%let ns="&ns";
%end;

proc transpose data=_&domain out=sp_&domain name=qnam label=qlabel;
  by studyid usubjid &idvar;
  var &crf &sponsor;
```

```
run;

data sp2_&domain;
   attrib
   RDOMAIN LABEL="Related Domain Abbreviation" FORMAT=$200. LENGTH=$200.
   IDVAR LABEL="Identifying Variable" FORMAT=$200. LENGTH=$200.
   IDVARVAL LABEL="Identifying Variable Value" FORMAT=$200. LENGTH=$200.
   QNAM LABEL="Qualifier Variable Name" FORMAT=$200. LENGTH=$200.
   QLABEL LABEL="Qualifier Variable Label" FORMAT=$200. LENGTH=$200.
   QVAL LABEL="Data Value" FORMAT=$200. LENGTH=$200.
   QORIG LABEL="Origin" FORMAT=$200. LENGTH=$200.
   QEVAL LABEL="Evaluator" FORMAT=$200. LENGTH=$200.
   ;
   SET SP_&domain;
   RDOMAIN=strip(upcase("&domain"));
   %if %length(&idvar)^=0 %then %do;
   IDVAR=strip(upcase("&idvar"));
   IDVARVAL=LEFT(PUT(&idvar,BEST.));
   %end;
   %else %do;
   idvar=' ';
   idvarval=' ';
   %end;
   QVAL=COL1;
    %if %length(&crf)^=0 %then %do;
   if strip(qnam) in (&ncrf) then do;
      qorig='CRF';
      qeval='  ';
   end;
   %end;
    %if %length(&sponsor)^=0 %then %do;
   if strip(qnam) in (&ns) then do;
      QORIG="DERIVED";
      QEVAL="SPONSOR";
   end;
  %end;
run;

%mformat(data=sp2_&domain);

PROC SQL;
  CREATE TABLE SUPP&domain (LABEL="Supplemental Qualifiers for &domain")
  AS
  SELECT
  STUDYID, RDOMAIN, USUBJID, IDVAR, IDVARVAL, QNAM, QLABEL, QVAL, QORIG, QEVAL
  FROM Sp2_&domain
  where qval^=' ';
QUIT;
%mend;
```

MACRO %CO

```
%macro co;
ods listing close;
ods output Variables=var;
 proc contents data=&protlib.p._all_;
run;
ods listing;

proc sql noprint;
   select count(distinct member) into: tot
   from var
   where index(variable, 'COVAL');
   /*obtain data having variable COVAL*/
   select distinct scan(member, 2,'.') into : mem1-:mem%left(&tot)
   from var
   where index(variable, 'COVAL');

   /*find the number of variable COVAL, COVAL1...*/
   select count(distinct variable) into: cov
   from var
   where index(variable, 'COVAL');
quit;

%do i=1 %to &tot;
   %let seq=%left(&&mem&i)seq;
   /*spefify the comment reference*/
   %let ref=%left(&&mem&i)spid;
   /*comments date*/
   %let dt=%left(&&mem&i)dtc;
   %let dt2=%left(&&mem&i)STdtc;

   data _co&i;
      length idvar rdomain coref coref codtc $200;
      set &protlib.p.&&mem&i;
      idvar="%left(&&mem&i)SEQ";
      idvarval=strip(put(&seq,best.));
      rdomain=strip("&&mem&i");
      coref=&ref ;
      coref=scan(coref, 1, ' ');
      %if &&mem&i=EX %THEN %DO;
      CODTC=&DT2;
      %END;
      %else %do;
      codtc=&dt;
      %end;
    keep studyid USUBJID rdomain idvar idvarval coref codtc coval:  ;
  run;

  %if &i=1 %then %do;
     data sdtmco;
        set _co&i;
     run;
  %end;
  %else %do;
     data sdtmco;
       set sdtmco _co&i;
     run;
  %end;
%end;

%let cov=&cov;
%let cot=%eval(&cov-1);
data sdtmco2;
   set sdtmco;
   DOMAIN='CO';
   where coval^=' '
   %if cot>=1 %then %do;
      %do j=1 %to &cot;
        or coval&j^=" "
      %end;
       ;
   %end;
run;
```

```
%if &cot>=1 %then %do;
 %do k=1 %to &cot;

   %let var=;
 /*drop COVAL with all missing values*/
  data a;
     set sdtmco2 end=last;
     retain y 0;
     if coval&k^=' ' then y=1;
     if last then do;
       if y=0 then call symput('var', "coval&k");
     end;
  run;

  %if &var^=  %then %do;
   data sdtmco3;
     set sdtmco2;
     drop &var;
   run;
  %end;
 %end;
%end;

proc sql noprint;
   select count(*) into: tot
   from sdtmco3;
quit;

%if &tot>0 %then %do;
data _co;
  attrib
  STUDYID LABEL="Study Identifier"   FORMAT=$200.    LENGTH=$200
  DOMAIN LABEL="Domain Abbreviation"   FORMAT=$200.    LENGTH=$200
  RDOMAIN LABEL="Related Domain Abbreviation"   FORMAT=$200.    LENGTH=$200
  USUBJID LABEL="Unique Subject Identifier"   FORMAT=$200.    LENGTH=$200
  COSEQ LABEL="Sequence Number"
  IDVAR LABEL="Identifying Variable "   FORMAT=$200.    LENGTH=$200
  IDVARVAL LABEL="Identifying Variable Value"   FORMAT=$200.    LENGTH=$200
  COREF LABEL="Comment Reference"   FORMAT=$200.    LENGTH=$200
  COVAL LABEL="Comment"   FORMAT=$200.    LENGTH=$200
  CODTC LABEL="Date/Time of Comment "   FORMAT=$200.    LENGTH=$200
   ;
   set sdtmco3;
   coseq=.;
run;
  %mseq(din=_co, dout=_co2, byvar=USUBJID IDVAR IDVARVAL, SEQKEY=USUBJID,VAR=COSEQ)
;
  %mformat(data=_co2);
  data CO(label="Comments");
     set _co2;
  run;
%end;
%mend;
```

REFERENCES

[1] SDTM Implementation Guide for Human Clinical Trials (SDTMIG v.3.1.2).

[2] Barry R. Cohen. SDTM, Plus or Minus. PharmaSUG, 2008

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Suwen Li
Everest Clinical Research Services Inc.
675 Cochrane Drive
Suite 408, East Tower
Markham, Ontario, Canada L3R 0B8
(905) 752-5253
suwen.li@ecrscorp.com

sas® Certified Advanced Programmer