PharmaSUG China 2018 – Paper CD-03

Knock Knock!!! Who's There???
Challenges faced while pooling data and studies for FDA submission

Amit Baid, CLINPROBE LLC, Acworth, GA USA

ABSTRACT

Pooling studies is not new to the pharmaceutical world. Almost every pharma company conducts clinical trials and goes through FDA submission for its drug to be released in the market. There are 'n' number of individual studies that are not powered to identify trends and rare adverse events. For a drug to be considered safe by FDA these individual studies need to be pooled to get a better picture of rare adverse events. Pooling data across multiple studies need to have the same structure and metadata standards and most of the time it is challenging as there might be legacy studies which didn't follow CDISC standards as compared to the newer studies. This paper will look at all the challenges and issues that we face while pooling data and studies together for a successful FDA submission and provide tips on how to handle them with ease through careful observation and planning.

INTRODUCTION

Every year FDA receives tons of NDA application from different pharmaceutical companies - consisting of multiple studies pooled together providing a better understanding of the safety profile of the drug. The data could be analyzed to see how the drug is behaving and if there are any rare adverse events which might result in the rejection of the drug being submitted. Since there are different studies from different phases of clinical trials it sometimes becomes challenging to pool them and careful planning is needed to integrate the studies. More and more studies are nowadays becoming CDISC compliant. This is beneficial to someone working on pooling different studies to create an integrated database as less time is spent standardizing the data from different studies and more time could be spent on critical analysis.

The main objective of this paper is to provide programming guidelines with examples to deal with the challenges faced while pooling data from different studies.

PLANNING

For any integration, whether for a submission or not, a plan should be developed as early as possible. The planning should involve the inclusion of studies and data for pooling from the perspective of both safety and efficacy analysis. Key team members such as Biostats, Data Management, Regulatory Affairs, Drug Safety and Medical Writers should be involved in planning. Proper timelines, resources and responsibilities need to be discussed and agreed for a successful execution of integration.

Pooling from ADaM

Pooling can be done either using SDTM (raw data) or ADaM (derived data) or both. In many cases teams may prefer to use derived data if these are already available at the study level and provided the studies share a common data standard. This is of relevance for pooling of efficacy data which often includes a large number of derived endpoints. Also using derived data for pooling can reduce our work to rederive common endpoints.

However proper care should be taken to ensure that all derivations are consistent across individual studies.

Pooling from SDTM

Pooling from raw data may be a suitable option when there is a significant variability in data structures, standards or endpoint derivations across studies. The usage of raw data will require reprogramming of endpoints, variables and other derivations already done at a study level. This is necessary if endpoints in the final pooling are different from the ones defined in one or more studies. The advantage of using raw data is that we can maximize consistency across trials if endpoints, baseline definitions, visit windowing, etc. are rederived.

Pooling from both SDTM and ADaM

In some cases, we might pool studies using both raw and derived datasets. This could happen when some studies have the required endpoints defined in the derived datasets while others don't. Also, this could happen when in the most recent or ongoing study the corresponding derived datasets are not yet available. One more scenario in which this could happen is that in derived datasets not all patients are included, but for specific analysis we might need all patients.

GETTING STARTED

It is always a challenging task when you deal with large number of studies to create an integrated database and it is very easy to get confused dealing with so many studies. CDISC has defined pretty good standards and if the studies are already in SDTM format, work involved in integration is much easier. SDTM datasets are easier to pool as they are mapped from the raw data and don't have analysis variables. In other words, they are like raw data. Integrated ADaM can be created from an integrated SDTM database.

Before we create an integrated SDTM database the following checks needs to be performed:

1) Domain Check

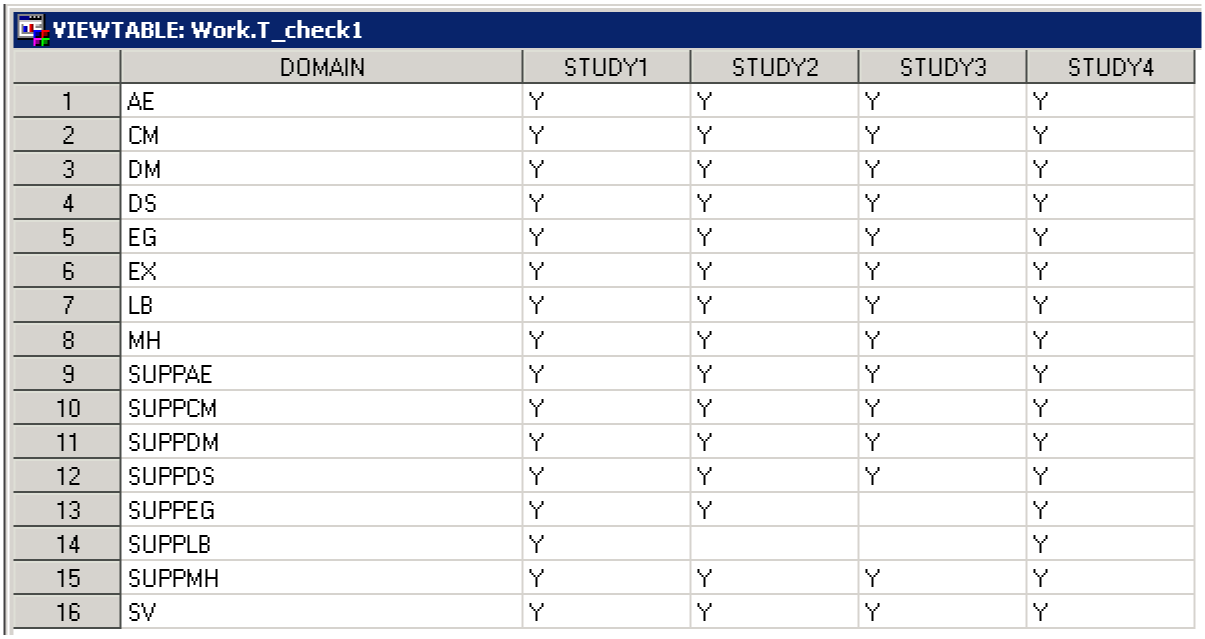
Before we create an integrated SDTM database it is recommended to get an overview of domains in all the studies. We should perform a domain check across all the studies to make sure they exist. This will help us avoid any issues in programming. Based on the study design there might be a supplemental qualifier domain in one study and not in other. We can merge the supplemental domain with the parent domain for creating an integrated ADaM.

The following code will produce a dataset (Table 1) with an overview of domains across different studies.

```
PROC SQL;
  CREATE TABLE CHECK1 AS
  SELECT MEMNAME AS DOMAIN,
         LIBNAME, "Y" AS STUDY
  FROM DICTIONARY.MEMBERS
  WHERE MEMTYPE="DATA" AND LIBNAME IN ("STUDY1","STUDY2","STUDY3","STUDY4")
  ORDER BY DOMAIN, LIBNAME;
QUIT;

PROC TRANSPOSE DATA=CHECK1 OUT=T_CHECK1(DROP=_NAME_);
  BY DOMAIN;
  ID LIBNAME;
  VAR STUDY;
RUN;
```

Table 1: Overview of Domain across all the studies



VIEWTABLE: Work.T_check1

	DOMAIN	STUDY1	STUDY2	STUDY3	STUDY4
1	AE	Y	Y	Y	Y
2	CM	Y	Y	Y	Y
3	DM	Y	Y	Y	Y
4	DS	Y	Y	Y	Y
5	EG	Y	Y	Y	Y
6	EX	Y	Y	Y	Y
7	LB	Y	Y	Y	Y
8	MH	Y	Y	Y	Y
9	SUPPAE	Y	Y	Y	Y
10	SUPPCM	Y	Y	Y	Y
11	SUPPDM	Y	Y	Y	Y
12	SUPPDS	Y	Y	Y	Y
13	SUPPEG	Y	Y		Y
14	SUPPLB	Y			Y
15	SUPPMH	Y	Y	Y	Y
16	SV	Y	Y	Y	Y

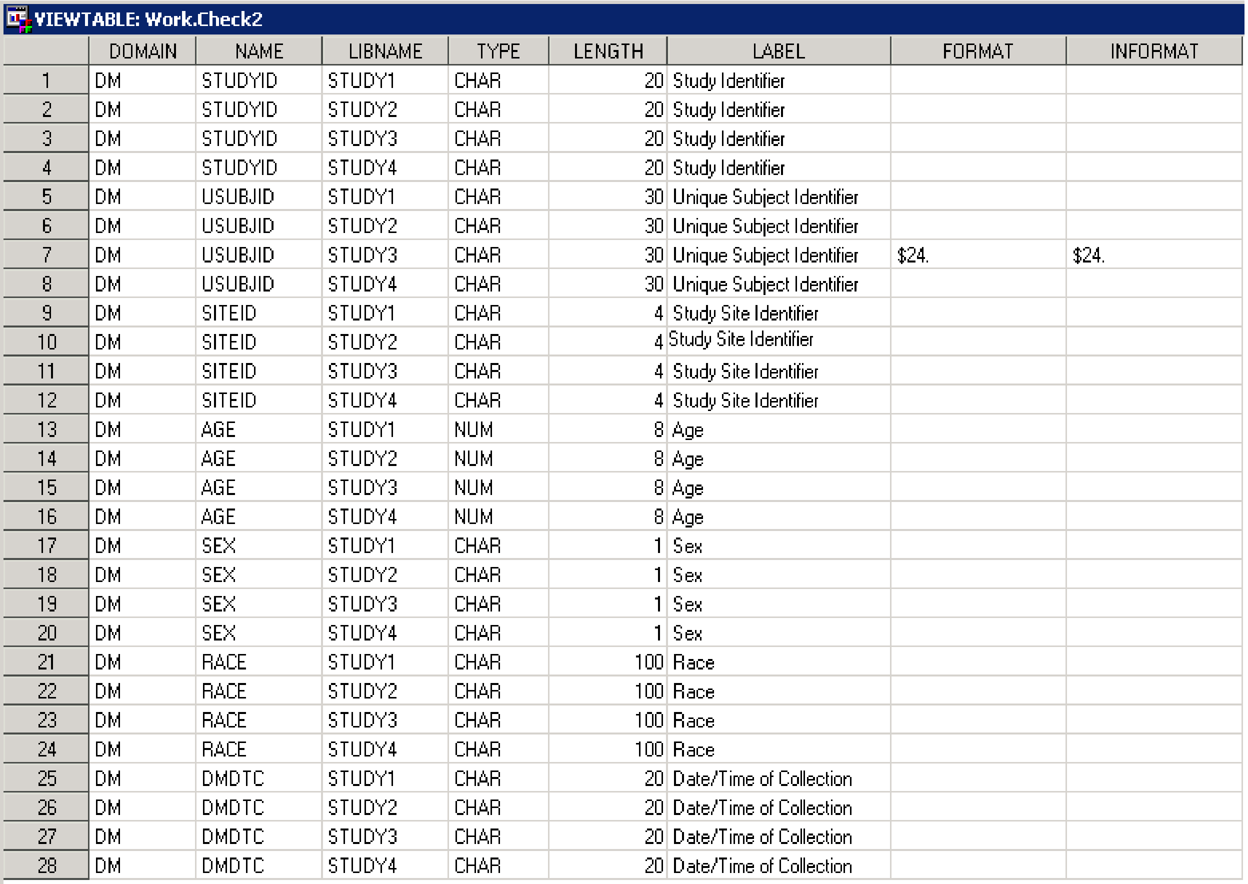
2) Attribute Check

It is important to check the variable attributes of a domain across all the studies. This gives us a clear picture of the data whether there are any attribute differences among studies or not. Based on the differences we need to adjust our programs to make the variable attributes uniform across all the studies. The attributes check can be performed in SAS using PROC CONTENTS of the same domain across all the studies and then stacking the datasets by NAME. We can then compare the TYPE, LENGTH, LABEL, FORMAT and INFORMAT for all the studies. We can also use PROC COMPARE to compare the results, but the only limitation in using this procedure is that we can compare only 2 datasets at one time.

The following code in PROC SQL makes our task much simpler and produces a dataset (Table 2) with an overview of variable attributes across different studies.

```
PROC SQL;
  CREATE TABLE CHECK2 AS
  SELECT MEMNAME AS DOMAIN,
         NAME AS NAME,
         LIBNAME AS LIBNAME,
         UPCASE(TYPE) AS TYPE,
         LENGTH AS LENGTH,
         LABEL AS LABEL,
         FORMAT AS FORMAT,
         INFORMAT AS INFORMAT
  FROM DICTIONARY.COLUMNS
  WHERE LIBNAME IN ("STUDY1","STUDY2","STUDY3","STUDY4") AND
        MEMTYPE="DATA" AND MEMNAME="DM"
  ORDER BY DOMAIN, VARNUM, NAME, LIBNAME;
QUIT;
```

Table 2: Overview of Variable Attributes in DM domain across studies



	DOMAIN	NAME	LIBNAME	TYPE	LENGTH	LABEL	FORMAT	INFORMAT
1	DM	STUDYID	STUDY1	CHAR	20	Study Identifier		
2	DM	STUDYID	STUDY2	CHAR	20	Study Identifier		
3	DM	STUDYID	STUDY3	CHAR	20	Study Identifier		
4	DM	STUDYID	STUDY4	CHAR	20	Study Identifier		
5	DM	USUBJID	STUDY1	CHAR	30	Unique Subject Identifier		
6	DM	USUBJID	STUDY2	CHAR	30	Unique Subject Identifier		
7	DM	USUBJID	STUDY3	CHAR	30	Unique Subject Identifier	$24.	$24.
8	DM	USUBJID	STUDY4	CHAR	30	Unique Subject Identifier		
9	DM	SITEID	STUDY1	CHAR	4	Study Site Identifier		
10	DM	SITEID	STUDY2	CHAR	4	Study Site Identifier		
11	DM	SITEID	STUDY3	CHAR	4	Study Site Identifier		
12	DM	SITEID	STUDY4	CHAR	4	Study Site Identifier		
13	DM	AGE	STUDY1	NUM	8	Age		
14	DM	AGE	STUDY2	NUM	8	Age		
15	DM	AGE	STUDY3	NUM	8	Age		
16	DM	AGE	STUDY4	NUM	8	Age		
17	DM	SEX	STUDY1	CHAR	1	Sex		
18	DM	SEX	STUDY2	CHAR	1	Sex		
19	DM	SEX	STUDY3	CHAR	1	Sex		
20	DM	SEX	STUDY4	CHAR	1	Sex		
21	DM	RACE	STUDY1	CHAR	100	Race		
22	DM	RACE	STUDY2	CHAR	100	Race		
23	DM	RACE	STUDY3	CHAR	100	Race		
24	DM	RACE	STUDY4	CHAR	100	Race		
25	DM	DMDTC	STUDY1	CHAR	20	Date/Time of Collection		
26	DM	DMDTC	STUDY2	CHAR	20	Date/Time of Collection		
27	DM	DMDTC	STUDY3	CHAR	20	Date/Time of Collection		
28	DM	DMDTC	STUDY4	CHAR	20	Date/Time of Collection		

By carefully observing the table above for differences we can adjust the length and other attributes before integrating the DM domain.

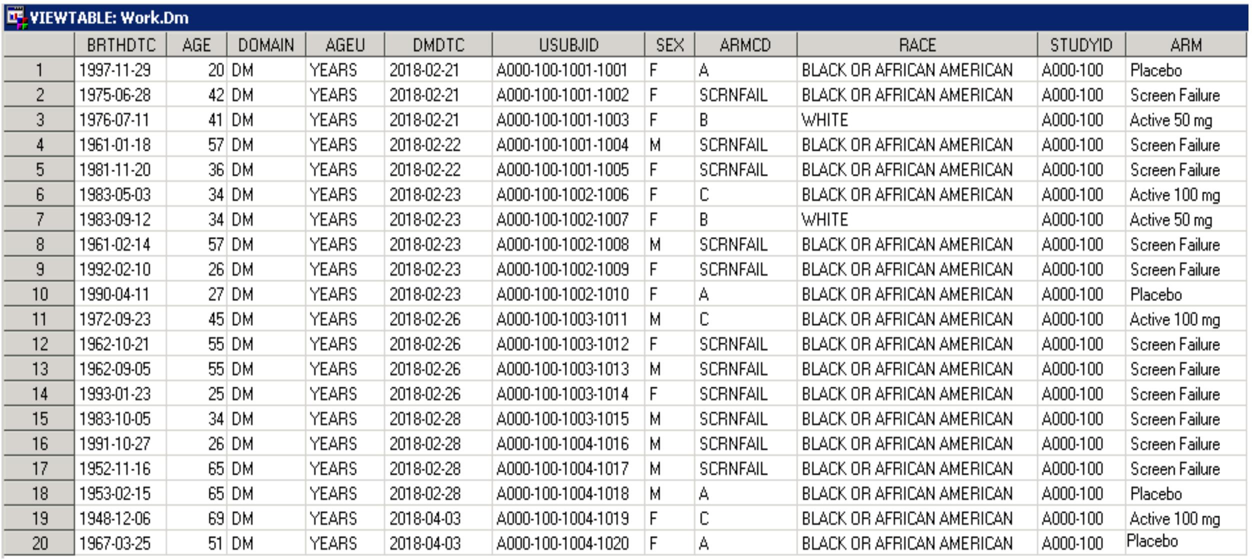
3) Ordering of Variables

Whenever we work on a dataset creation, whether it is for an individual study or an integrated database the most important thing is the order in which the variables appear in the dataset. Ordering facilitates easy viewing and gives us a clear understanding of the domain. For example, every domain starts with STUDYID, USUBJID, SUBJID, SITEID and so on. You don’t want to see STUDYID or USUBJID towards the very end.

For this to work, we need to have data specifications in place. We can follow the CDISC SDTM or ADAM implementation guide for the ordering. Normally the first group of variables are the identification variables such as STUDYID, USUBJID, SUBJID, SITEID, etc., followed by the demographic variables such as AGE, SEX, RACE, ETHNIC, etc.

For example, we have the following table (Table 3A). Please note the ordering is not correct as STUDYID is towards the end and USUBJID is somewhere in the middle. This can be fixed by setting the order as per the data specification (Table 3B).

Table 3A



	BRTHDTC	AGE	DOMAIN	AGEU	DMDTC	USUBJID	SEX	ARMCD	RACE	STUDYID	ARM
1	1997-11-29	20	DM	YEARS	2018-02-21	A000-100-1001-1001	F	A	BLACK OR AFRICAN AMERICAN	A000-100	Placebo
2	1975-06-28	42	DM	YEARS	2018-02-21	A000-100-1001-1002	F	SCRNFAIL	BLACK OR AFRICAN AMERICAN	A000-100	Screen Failure
3	1976-07-11	41	DM	YEARS	2018-02-21	A000-100-1001-1003	F	B	WHITE	A000-100	Active 50 mg
4	1961-01-18	57	DM	YEARS	2018-02-22	A000-100-1001-1004	M	SCRNFAIL	BLACK OR AFRICAN AMERICAN	A000-100	Screen Failure
5	1981-11-20	36	DM	YEARS	2018-02-22	A000-100-1001-1005	F	SCRNFAIL	BLACK OR AFRICAN AMERICAN	A000-100	Screen Failure
6	1983-05-03	34	DM	YEARS	2018-02-23	A000-100-1002-1006	F	C	BLACK OR AFRICAN AMERICAN	A000-100	Active 100 mg
7	1983-09-12	34	DM	YEARS	2018-02-23	A000-100-1002-1007	F	B	WHITE	A000-100	Active 50 mg
8	1961-02-14	57	DM	YEARS	2018-02-23	A000-100-1002-1008	M	SCRNFAIL	BLACK OR AFRICAN AMERICAN	A000-100	Screen Failure
9	1992-02-10	26	DM	YEARS	2018-02-23	A000-100-1002-1009	F	SCRNFAIL	BLACK OR AFRICAN AMERICAN	A000-100	Screen Failure
10	1990-04-11	27	DM	YEARS	2018-02-23	A000-100-1002-1010	F	A	BLACK OR AFRICAN AMERICAN	A000-100	Placebo
11	1972-09-23	45	DM	YEARS	2018-02-26	A000-100-1003-1011	M	C	BLACK OR AFRICAN AMERICAN	A000-100	Active 100 mg
12	1962-10-21	55	DM	YEARS	2018-02-26	A000-100-1003-1012	F	SCRNFAIL	BLACK OR AFRICAN AMERICAN	A000-100	Screen Failure
13	1962-09-05	55	DM	YEARS	2018-02-26	A000-100-1003-1013	M	SCRNFAIL	BLACK OR AFRICAN AMERICAN	A000-100	Screen Failure
14	1993-01-23	25	DM	YEARS	2018-02-26	A000-100-1003-1014	F	SCRNFAIL	BLACK OR AFRICAN AMERICAN	A000-100	Screen Failure
15	1983-10-05	34	DM	YEARS	2018-02-28	A000-100-1003-1015	M	SCRNFAIL	BLACK OR AFRICAN AMERICAN	A000-100	Screen Failure
16	1991-10-27	26	DM	YEARS	2018-02-28	A000-100-1004-1016	M	SCRNFAIL	BLACK OR AFRICAN AMERICAN	A000-100	Screen Failure
17	1952-11-16	65	DM	YEARS	2018-02-28	A000-100-1004-1017	M	SCRNFAIL	BLACK OR AFRICAN AMERICAN	A000-100	Screen Failure
18	1953-02-15	65	DM	YEARS	2018-02-28	A000-100-1004-1018	M	A	BLACK OR AFRICAN AMERICAN	A000-100	Placebo
19	1948-12-06	69	DM	YEARS	2018-04-03	A000-100-1004-1019	F	C	BLACK OR AFRICAN AMERICAN	A000-100	Active 100 mg
20	1967-03-25	51	DM	YEARS	2018-04-03	A000-100-1004-1020	F	A	BLACK OR AFRICAN AMERICAN	A000-100	Placebo

Table 3B

Variable Name	Variable Label	Type	Length
STUDYID	Study Identifier	Char	$20
DOMAIN	Domain Abbreviation	Char	$2
USUBJID	Unique Subject Identifier	Char	$30
BRTHDTC	Date/Time of Birth	Char	$20
AGE	Age	Num	
AGEU	Age Units	Char	$6
SEX	Sex	Char	$1
RACE	Race	Char	$100
ARMCD	Planned Arm Code	Char	$20
ARM	Description of Planned Arm	Char	$100
DMDTC	Date/Time of Collection	Char	$20

Method 1:

We can set the order by using a RETAIN statement followed by the variable names in order. The code is given below:

```
DATA DM;
  RETAIN STUDYID DOMAIN USUBJID BRTHDTC AGE AGEU
         SEX RACE ARMCD ARM DMDTC;
  SET DM;
RUN;
```

Method 2:

In case of lot of variables, a macro could be developed which can read the data specifications for a domain and create an ordered variable list using PROC SQL. We can import the data specification workbook for a domain using PROC IMPORT.

We can define a name range by selecting the cells in EXCEL and holding CTRL+F3 and Clicking on "New" and typing a name. We can then use the RANGE="NameRange" option in PROC IMPORT. In the following example you will see DMRANGE is defined by selecting the cells needed from STUDYID through DMDTC.

Please note that this works for XLS format workbook only. If you have a workbook in XLSX format, then you might have to convert it to XLS format.

```
PROC IMPORT
  DATAFILE="A000-100-SDTM Domain Mapping Specifications.xls"
  OUT=SPECS
  DBMS=XLS
  REPLACE;
  GETNAMES=NO;
  SHEET="DM";
  RANGE="DMRANGE";
  DATAROW=2;
RUN;
```

The following PROC SQL code will create an ordered variable list and will store it in a macro variable VARLIST which can be used in a RETAIN statement in a data step given below. It is a good idea to use the RETAIN statement in a separate data step after all the variables have been created.

```
PROC SQL NOPRINT;
  SELECT A
  INTO: VARLIST
  SEPARATED BY " "
  FROM SPECS;
QUIT;
```

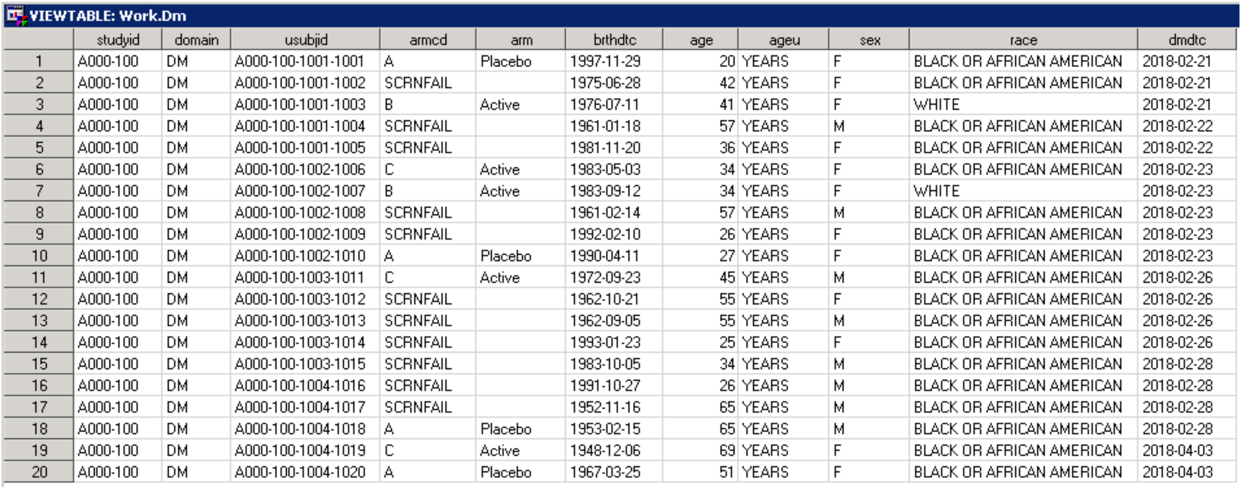
```
DATA CHECK3;
  RETAIN &VARLIST;
  SET DM;
RUN;
```

4) Variable Names in UPCASE

The industry wide standard for any SDTM or ADAM dataset is to have all the variables in UPPERCASE for easy readability. Sometimes a programmer can accidentally create the final variables in LOWERCASE. To avoid this, we can write a small macro which will change the case of the variable and make it UPPERCASE.

In the below table (Table 4) you will see that the variable names are in lowercase.

Table 4



VIEWTABLE: Work.Dm

	studyid	domain	usubjid	armcd	arm	brthdtc	age	ageu	sex	race	dmdtc
1	A000-100	DM	A000-100-1001-1001	A	Placebo	1997-11-29	20	YEARS	F	BLACK OR AFRICAN AMERICAN	2018-02-21
2	A000-100	DM	A000-100-1001-1002	SCRNFAIL		1975-06-28	42	YEARS	F	BLACK OR AFRICAN AMERICAN	2018-02-21
3	A000-100	DM	A000-100-1001-1003	B	Active	1976-07-11	41	YEARS	F	WHITE	2018-02-21
4	A000-100	DM	A000-100-1001-1004	SCRNFAIL		1961-01-18	57	YEARS	M	BLACK OR AFRICAN AMERICAN	2018-02-22
5	A000-100	DM	A000-100-1001-1005	SCRNFAIL		1981-11-20	36	YEARS	F	BLACK OR AFRICAN AMERICAN	2018-02-22
6	A000-100	DM	A000-100-1002-1006	C	Active	1983-05-03	34	YEARS	F	BLACK OR AFRICAN AMERICAN	2018-02-23
7	A000-100	DM	A000-100-1002-1007	B	Active	1983-09-12	34	YEARS	F	WHITE	2018-02-23
8	A000-100	DM	A000-100-1002-1008	SCRNFAIL		1961-02-14	57	YEARS	M	BLACK OR AFRICAN AMERICAN	2018-02-23
9	A000-100	DM	A000-100-1002-1009	SCRNFAIL		1992-02-10	26	YEARS	F	BLACK OR AFRICAN AMERICAN	2018-02-23
10	A000-100	DM	A000-100-1002-1010	A	Placebo	1990-04-11	27	YEARS	F	BLACK OR AFRICAN AMERICAN	2018-02-23
11	A000-100	DM	A000-100-1003-1011	C	Active	1972-09-23	45	YEARS	M	BLACK OR AFRICAN AMERICAN	2018-02-26
12	A000-100	DM	A000-100-1003-1012	SCRNFAIL		1962-10-21	55	YEARS	F	BLACK OR AFRICAN AMERICAN	2018-02-26
13	A000-100	DM	A000-100-1003-1013	SCRNFAIL		1962-09-05	55	YEARS	M	BLACK OR AFRICAN AMERICAN	2018-02-26
14	A000-100	DM	A000-100-1003-1014	SCRNFAIL		1993-01-23	25	YEARS	F	BLACK OR AFRICAN AMERICAN	2018-02-26
15	A000-100	DM	A000-100-1003-1015	SCRNFAIL		1983-10-05	34	YEARS	M	BLACK OR AFRICAN AMERICAN	2018-02-28
16	A000-100	DM	A000-100-1004-1016	SCRNFAIL		1991-10-27	26	YEARS	M	BLACK OR AFRICAN AMERICAN	2018-02-28
17	A000-100	DM	A000-100-1004-1017	SCRNFAIL		1952-11-16	65	YEARS	M	BLACK OR AFRICAN AMERICAN	2018-02-28
18	A000-100	DM	A000-100-1004-1018	A	Placebo	1953-02-15	65	YEARS	M	BLACK OR AFRICAN AMERICAN	2018-02-28
19	A000-100	DM	A000-100-1004-1019	C	Active	1948-12-06	69	YEARS	F	BLACK OR AFRICAN AMERICAN	2018-04-03
20	A000-100	DM	A000-100-1004-1020	A	Placebo	1967-03-25	51	YEARS	F	BLACK OR AFRICAN AMERICAN	2018-04-03

The following code could be used to change all the variable names to UPPERCASE in the dataset.

Once you have the variable list in a macro as given in the above example, we can use the following to make the variable names appear in UPPERCASE.

```
DATA CHECK3;
  RETAIN %UPCASE(&VARLIST);
  SET DM;
RUN;
```

5) Length Trimming

For FDA submission the SDTM and ADaM datasets must be in an approved format, otherwise they cannot be used and reviewed. The SAS Transport Format (XPORT) Version 5 is the file format for the submission of all electronic datasets. The XPORT is an open file format published by SAS Institute for the exchange of study data. Data can be translated to and from XPORT to other commonly used formats without the use of SAS programs or any other vendor specific programs. There should be one dataset per transport file and the dataset in the transport file should be named the same as the transport file (e.g. AE and AE.XPT). XPORT files can be created by using the COPY Procedure in SAS Version 5 and higher. However, there are some requirements before converting a SAS dataset to XPT format which are as follows:

- Dataset and variable names must be up to 8 characters
- Dataset and variable labels must be up to 40 characters
- Dataset and variable names and labels should only include ASCII (American Standard Code for Information Interchange) text codes
- Character variables must be up to 200 characters in length

Before submitting the datasets in XPT format we need to make sure they are not big enough. To meet these requirements:

- Character variables in main domains need to be trimmed to the minimum length needed across datasets
- Character variables in supplemental domains need to be trimmed to the minimum length needed within each dataset.

A simple macro to trim the length is given below:

```
%MACRO TRIM_LENGTH (LIB=, DSN=);

PROC SQL;
  CREATE TABLE CHECK4A AS
  SELECT UPCASE(NAME) AS NAME,
         LENGTH AS O_LENGTH,
         VARNUM AS VARNUM
  FROM DICTIONARY.COLUMNS
  WHERE UPCASE(LIBNAME)="&LIB" AND UPCASE(MEMTYPE)="DATA" and
        UPCASE(MEMNAME)="&DSN" AND UPCASE(TYPE)="CHAR"
  ORDER BY VARNUM;
QUIT;

DATA _NULL_;
  SET CHECK4A END=EOF;
  CALL SYMPUT("VAR"||STRIP(PUT(_N_,8.)),STRIP(NAME));
  IF EOF THEN CALL SYMPUT("TOTVAR",STRIP(PUT(_N_,8.)));
RUN;

PROC SQL;
  CREATE TABLE CHECK4B AS
  %DO I=1 %TO %EVAL(&TOTVAR-1);
  SELECT "&&VAR&I" AS NAME, MAX(LENGTH(&&VAR&I)) AS N_LENGTH
  FROM &LIB..&DSN
  UNION
  %END;
  SELECT "&&VAR&TOTVAR" AS NAME, MAX(LENGTH(&&VAR&TOTVAR)) AS N_LENGTH
  FROM &LIB..&DSN;
QUIT;

PROC SQL;
  CREATE TABLE CHECK4 AS
  SELECT A.*, B.N_LENGTH
  FROM CHECK4A AS A, CHECK4B AS B
  WHERE A.NAME EQ B.NAME AND A.O_LENGTH NE B.N_LENGTH
  ORDER BY VARNUM;
QUIT;

PROC SQL NOPRINT;
  SELECT STRIP(NAME)||" $"||STRIP(PUT(N_LENGTH,8.)) INTO: SETLEN
  SEPARATED BY " "
  FROM CHECK4;
QUIT;

OPTIONS VARLENCHK=NOWARN;

DATA &DSN;
  LENGTH &SETLEN;
  SET &LIB..&DSN;
RUN;

%MEND TRIM_LENGTH;

%TRIM_LENGTH (LIB=STUDY1, DSN=DS);
```

6) MedDRA Version Levelling

MedDRA stands for Medical Dictionary for Regulatory Activities. It is a clinically validated international medical terminology dictionary used by regulatory authorities in the pharmaceutical industry from pre-marketing to post-marketing activities, and for data entry, retrieval, evaluation, and presentation. In addition, it is the adverse event classification dictionary endorsed by the International Conference on Harmonization (ICH). MedDRA is widely used internationally, including in the United States, European Union, and Japan. Its use is currently mandated in Europe and Japan for safety reporting.

The MedDRA dictionary is organized by System Organ Class (SOC), divided into High-Level Group Terms (HLGT), High-Level Terms (HLT), Preferred Terms (PT) and finally into Lowest Level Terms (LLT). In addition, the MedDRA dictionary includes Standardized MedDRA Queries (SMQs). SMQs are grouping of terms that relate to a defined medical condition or area of interest.

Individual cases are usually coded for data entry at the most specific (LLT) level, and outputs of counts or cases are usually provided at the PT level. The higher levels (HLT, HLGT and SOC) as well as SMQ are used for searching and for organization and subtotaling of outputs.

Updated MedDRA versions are released twice a year – in March and September. The March release is the main annual release and contains changes at the HLT level and above along with LLT and PT changes. The September release typically contains changes only at the LLT and PT level. The March 2018 Version 21.0 release is the current version.

Coming back to our pooling of studies, it is very likely that different studies used different MedDRA versions for coding adverse events. It is very important to code all adverse events from all the studies using same version of MedDRA to avoid differences and to correctly identify number of adverse events. We can do the MedDRA levelling in individual studies before pooling the data or after creating an integrated SDTM AE domain.

7) Controlled Terminology

It is very likely that different studies used different controlled terminology within the values of the SDTM variables in comparison to CDISC SDTM controlled terminology. Differences could be because of not following SDTM controlled terminology as intended. It is advisable to map any differences to the latest available CDISC SDTM controlled terminology.

Please see the table below for example:

Variable	Study 1	Study 2	SDTM Controlled Terminology
SEX	M	MALE	M
	F	FEMALE	F
COUNTRY	AUSTRIA	AUS	AUS
	CANADA	CAN	CAN
AESER	Y	YES	Y
	N	NO	N
AEREL	Y	DEFINITELY RELATED	Y
	N	NOT RELATED	N
		POSSIBLY RELATED	
		PROBABLY RELATED	
		UNLIKELY RELATED	

8) Baseline Check

The baseline flags or values collected in some SDTMs are not defined consistently across studies. There might be a different definition of baseline for the ISS analysis. For a rollover study the baseline always comes from the parent study so we might have to drop it in the rollover study. For a crossover study there might be multiple baselines depending on the treatments and same is the case with studies with multiple periods.

For example, in a SDTM domain baseline might be defined as the last assessment before the study dose but for analysis we might need to take an average of all assessments prior to dosing.

It is always advisable to re-define baseline in the ADaM data as per the statistical analysis plan.

9) PARAMCD, PARAM, PARAMN mappings

In SDTM findings domain we map the XXTESTCD and XXTEST to PARAMCD and PARAM of ADaM respectively. The standard units XXSTRESU are always attached to the XXTEST within parentheses before assigning to PARAM. There might be a few tests, for which units were not collected such as tests in urinalysis panel of a laboratory data. So, in those cases there are no units being attached to XXTEST.

PARAMN is always defined after sorting PARAM alphabetically and assigning a unique parameter number starting with 1. Each PARAMCD should have a one-to-one mapping with PARAM. There cannot be multiple units for the same parameter.

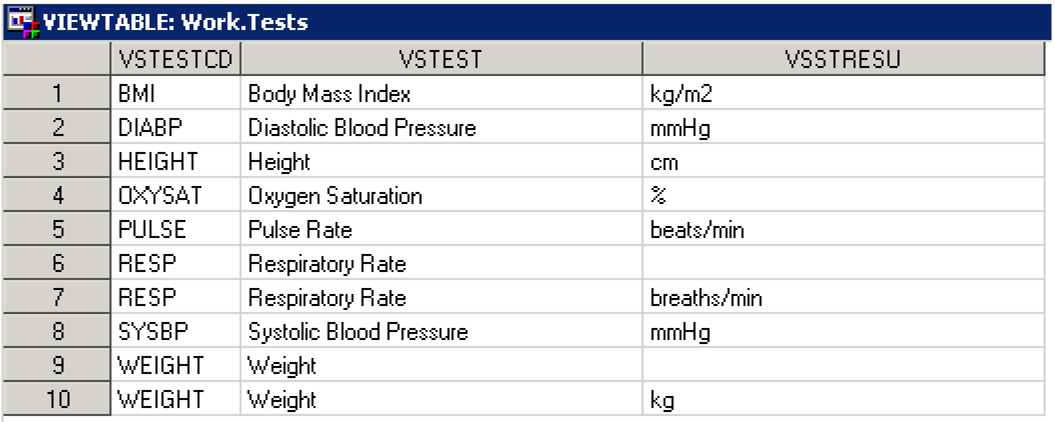
Units of various tests or parameters could be different across studies. For example, a laboratory data might be collected at different sites thus resulting in multiple units. It is always advisable to check units for all the tests and parameters being summarized for the analysis. They should be then adjusted as per the analysis. Normally when the collected value is missing the standard units will be missing in these cases. But to have a unique mapping these should be fixed.

The below code will help us gather all the combination of VSTESTCD, VSTEST and VSSTRESU from SDTM.VS and map them to PARAMCD and PARAM in ADAM.ADVS uniquely.

```
PROC SQL;
  CREATE TABLE VS AS
  SELECT *
  FROM STUDY1.VS
  ORDER BY VSTESTCD, VSTEST, USUBJID, VISITNUM, VSDTC;
QUIT;

PROC SQL;
  CREATE TABLE TESTS AS
  SELECT DISTINCT VSTESTCD, VSTEST, VSSTRESU
  FROM VS
  ORDER BY VSTESTCD, VSTEST;
QUIT;

DATA VS;
  MERGE VS(IN=A DROP=VSSTRESU) TESTS(IN=B WHERE=(VSSTRESU NE ""));
  BY VSTESTCD VSTEST;
RUN;
```


VIEWTABLE: Work.Tests

	VSTESTCD	VSTEST	VSSTRESU
1	BMI	Body Mass Index	kg/m2
2	DIABP	Diastolic Blood Pressure	mmHg
3	HEIGHT	Height	cm
4	OXYSAT	Oxygen Saturation	%
5	PULSE	Pulse Rate	beats/min
6	RESP	Respiratory Rate	
7	RESP	Respiratory Rate	breaths/min
8	SYSBP	Systolic Blood Pressure	mmHg
9	WEIGHT	Weight	
10	WEIGHT	Weight	kg

10) Date/Time Variable Check

The dates in the SDTM data are in character format which cannot be used for the analysis. For analysis purpose these dates should be converted to numeric. While creating ADaM it becomes necessary to determine whether there is a need for all three DATE, TIME and DATETIME variables or just DATE variables. If TIME is not being used for analysis, then we just need the DATE variable.

It is advisable to create one SAS macro/program to convert character date to numeric instead of handling it separately in each ADaM program. It is easy for maintenance as we can make changes in one program thus saving our time. If there are any partial or missing dates and we need to do any imputation based on the statistical analysis plan, then there should be corresponding imputation flag.

SCOPE OF ANALYSIS

Studies might be pooled by indication or therapeutic area and is primarily dependent on the deliverable and objective of the analysis. As discussed before pooling could be done using SDTM database. After the creation of integrated SDTM database, programmer could focus on creation of integrated ADaM for integrated summary of safety. Integrated summary of safety reports contains adverse events, demographics, deaths, discontinuation information, and extent of exposure and laboratory results. Other safety information like electrocardiogram, physical examination, vital signs, etc. may also be included based on the scope of analysis. During the planning process it is advisable to focus on analysis scope and determine how many ADaM datasets are needed. There might not be a need to have all the SDTM variables for the analysis. Some permissible SDTM variables could be dropped from the integrated SDTM domains if they are not required for the analysis.

VALIDATION

Validation is an important step in the process of integration. A proper validation plan is needed before the start of integration. After the integration of ADaM datasets, validation can be done by cross checking against the individual study data. The idea behind this approach is to make sure that no information is lost during integration of analysis data and whether the integrated analysis data are a true representation of individual study data. The integrated ADaM database can be subset for individual studies and reports can be created and compared with the individual study CSR. Any new derivation or adjustments in the integrated analysis datasets for controlled terminology, units, and treatment information could possibly give us differences while cross checking against the individual study CSR, thus allowing us to check the validity of the new integrated analysis dataset.

CONCLUSION

For any task to be done careful observation and planning are required. Handling data from different studies could be challenging and daunting but it can be made easier by following the CDISC standard guidelines and good programming practices. The examples and approach mentioned in this paper provides us with a proper solution for building an integrated database for submissions.

REFERENCES

- Analysis Data Model (ADaM) Implementation Guide, Version 1.0, Final, Published Dec 17, 2009
- Study Data Tabulation Model (SDTM) Implementation Guide, Version 3.1.2, Published Nov 12, 2008
- Study Data Technical Conformance Guide, Version 4.1, Published Mar 2018
- https://pharmasug.org/proceedings/2012/DS/PharmaSUG-2012-DS17.pdf
- https://www.lexjansen.com/nesug/nesug13/21_Final_Paper.pdf

ACKNOWLEDGEMENTS

I would like to thank Margaret Hung for inviting and encouraging me for conference participation. I would also like to thank Dilip Raghunathan of INSMED, Inc. and Sreedhar Bodepudi of Vertex Pharmaceuticals for consistently providing valuable guidance and support for the conference participation. I would also like to thank my other colleagues and friends particularly Ganesh Iyer and Kush Patel for providing review comments and last but not the least, my mom, dad and wife Ruby for all their support.

CONTACT INFORMATION

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

Amit Baid, M.S.
CLINPROBE, LLC
2525 Thorngate Drive
Acworth, GA 30101
Phone: +1 (619) 846-8842
Email: amit.baid@clinprobe.com