PharmaSUG 2016 - Paper TT18

# Duplicate records - it may be a good time to contact your data management team

Sergiy Sirichenko, Pinnacle 21, Plymouth Meeting, Pennsylvania
Max Kanevsky, Pinnacle 21, Plymouth Meeting, Pennsylvania

## ABSTRACT

Most programmers are already familiar with the concept of duplicate records, where multiple records are identical in values across all variables. These duplicates are easy to catch and clean. However, there are also cases where clinical data has more than one expected record for the same assessment at the same time point with different results. These are much harder to manage and can complicate analysis by producing incorrect outcomes. In this presentation, we will examine the concepts of the unique observation and key variables. We'll review common causes and examples of duplicate records during the data collection and mapping process. And finally we'll demonstrate how to detect, clean, and document duplicate records.

## INTRODUCTION

There are many definitions for duplicates. For example, “different or multiple records that refer to one unique real world entity or object” [1]. In this paper we define duplicate records as records with the same values in dataset Key Variables. For example, subject has two different records for the same test assessment at the same time point. In terms of the previous definition, a “real world entity” is defined by Key Variables.

High quality data is not expected to have duplicate records. Their presence is a sign of either a problem with data collection and insufficient data cleaning or an incorrect mapping to SDTM structure.

Recently FDA reported that most regulatory submissions have problems with duplicate records. They represent potentially contradictory information and make difficult to summarize results [2]. Therefore, FDA asks to remove duplicate records in submission data. However, at this point no additional details are provided on how to accomplish this and what are the FDA expectations?

## KEY VARIABLES

To understand a concept of duplicate records we need to start with Key Variables.

Key variables are intended to identify each record in a dataset. CDISC SDTM Implementation Guide uses two different types of Key Variables: “A *natural key* is a piece of data (one or more columns of an entity) that uniquely identify that entity, and distinguish it from any other row in the table. The advantage of natural keys is that they exist already… A *surrogate key* is a single-part, artificially established identifier for a record. Surrogate key assignment is a special case of derived data, one where a portion of the primary key is derived. A surrogate key is immune to changes in business needs. In addition, the key depends on only one field, so it‘s compact. A common way of deriving surrogate key values is to assign integer values sequentially. The --SEQ variable in the SDTM datasets is an example of a surrogate key for most datasets” [3] (#3.2.1.1).

Natural keys define a “real” data structure. However, in some cases, utilization of surrogate keys is also needed. The Trial Summary (TS) domain is a good example, when multiple records for the same parameter require the use of TSSEQ variable to uniquely identify a record.

## DIAGNOSTICS

There are two options to identify duplicate records. Using out-of-box (OOB) tools like Pinnacle 21 (formerly OpenCDISC) Validator [4], [5] is a good start. However, any OOB tools have some limitations and require additional tuning for company specific processes. To get more precise results utilization of custom programs driven by study specific metadata is expected.

We will use Pinnacle 21 (P21) Validator as an example for demonstrating the challenges in implementation of duplicate record checks.

P21 Community is a free open source application available for download on opencdisc.org or pinnacle21.net/community websites. P21 Enterprise is a commercial edition used by FDA (a.k.a. DataFit) and PMDA to assess standard compliance and data quality of regulatory submissions.

P21 Validator has a set of generic checks for duplicate records in Findings and Events domains. Only duplicate records are reported. The first ”*reference*” record is not included into validation output. Validation is performed as a

lookup based on pre-defined Key Variables. If any duplicate (not first) record is found, then it will be reported. Here is an example of P21 algorithm for AE domain in terms of SAS® code:

```
proc sort data=AE out=AE1;
    by USUBJID AETERM AEDECOD AESEV AESTDTC;
run;
data AEDUPS;
    set AE1;
    by USUBJID AETERM AEDECOD AESEV AESTDTC;
    if not first.AESTDTC then output;
    keep USUBJID AETERM AEDECOD AESEV AESTDTC;
run;
```

P21 Validator uses a Java based engine with validation specifications stored in XML format (as extension of Define-XML standard). These specification files are found in *pinnacle21-community\components\config* folder. Here is a definition of SD1201 rule for checking duplicates in Events domains

```
<val:Unique ID="SD1201" PublisherID="FDAC213"
Message="Duplicate records in %Domain% domain"
Description="The structure of Events class domains should be one records per Event
per subject. No Events with the same Collected Term (--TERM), Decoded Term (--
DECOD), Category (--CAT), Subcategory (--SCAT), Severity (--SEV), and Toxicity
Grade (--TOXGR) values for the same Subject (USUBJID) and the same Start Date (--
STDTC) are expected."
Category="Consistency"
Type="Warning"
Variable="%Domain%STDTC"
When="%Domain%TERM != &amp;apos;&amp;apos;"
GroupBy="USUBJID,%Domain%TERM,%Domain%DECOD,%Domain%CAT,%Domain%SCAT,%Domain%SEV,%D
omain%TOXGR"
Optional="%Domain%DECOD,%Domain%CAT,%Domain%SCAT,%Domain%SEV,%Domain%TOXGR"/>
```

This rule looks for *uniqueness* of --STDTC values group be USUBJID, --TERM, --DECOD, --CAT, --SCAT, --SEV, --TOXGR variables. The rule is compiled into a dataset specific check by the Validator engine. Variables listed in Optional attribute (--DECOD, --CAT, --SCAT, --SEV, --TOXGR) will be ignored if not present in a dataset. If other variables in a rule definition like USUBJID or --STDTC are not present in dataset, the check will not compile and will not execute. Such a generic algorithm allows the rule to adjust for different domains and particular implementations. For example, AETOXGR may be used instead of AESEV. Category and Subcategory are typical variables in Disposition, however they are quite rarely utilized in AE domain.

P21 algorithm for duplicate records in Findings domain (SD1117 rule) includes USUBJID, --TESTCD, --CAT, --SCAT, --METHOD, --SPEC, --LOC, --DRVFL, --EVAL, VISITNUM, --TPTNUM, --DTC variables.

Algorithms were adjusted across Validator versions to minimize false-positive (incorrectly reported) and false-negative (not reported) issue messages. Nevertheless, such generic algorithms cannot produce 100% accurate and correct results. For example, some domains use --ORRES or SUPQUAL info as key variables. For other domains, --ORRES can never be a key variable.

In theory, a define.xml file should be the source for study specific Key Variables for each dataset. However, incorrect or invalid Key Variables are too common an issue to utilize define.xml file for validation. Here are common examples

- Usage of --SEQ variables, which are *surrogate key* representing artificial identifier. Only *natural keys* (with only few exceptions) are expected to be used to define Key Variables in datasets
  - "*USUBJID, AESEQ*" – invalid metadata
  - "*USUBJID, AETERM, AESTDTC*" – expected metadata
- Usage of too many variables as Key Variables in dataset. Such approach does not correctly explain data structure. For example,
  - "*USUBJID, AETERM, AEDECOD, AELLT, AEHLT, AESOC, AESEV, AESER, AEREL, AESHOSP, AESTDTC, AEENDT, VISIT*"
- Usage of --REFID, --SPID variables without any details about them in define.xml file

- Usage of --SPID variable as artificial *surrogate key*. Such approach does not explain what is a source for duplicate records and how to analyze data. For example,
    - --SPID is a key variable with comment/derivation in define.xml *"--SPID variable was populate to ensure uniqueness of Key Variables".* This metadata is not much different from missing one.
    - Missing details on key variables in define.xml file. For example,
        - Sponsor provided an explanation in Reviewer's Guide for issue of duplicate records in PE domain: "*No actual duplicates exist as information included SUPPPE creates a unique sort*"
        - However a study define.xml file does not include any reference to SUPPPE dataset as a source of key variables: "*STUDYID, USUBJID, PETESTCD, VISITNUM, PEORRES*"

Therefore, Pinnacle 21 Validator can today only rely on generic Key Variables pre-defined as a part of check algorithms. In the future, when the industry can produce reliable high quality Define files, automated tools can fully utilize them as the source of study specific metadata.

The current P21 plan is to start introducing define.xml driven checks for duplicate records and refine generic checks by making them domain specific. For example, DA and LB domains should have separate sets of Key Variables.

Sponsors are responsible for data quality of their clinical trials. While OOB tools provide significant help, additional data cleaning is expected. There are always at least two drivers for continuous improvement for any existing diagnostic tool: usability and precision. SAS programmers can handle implementation of study specific checks for duplicates. While structure and content of validation reports may be adjusted for particular needs. For example, adding the first "reference" record and including all variables for better diagnostics for duplicate records may improve standard P21 reports. An example SAS code for AE duplicates above may be modified as follow:

```
proc sort data=AE out=AE1;
    by USUBJID AETERM AEDECOD AESEV AESTDTC;
run;
data AEDUPS;
    set AE1;
    by USUBJID AETERM AEDECOD AESEV AESTDTC;
    if not first.AESTDTC or not last.AESTDTC then output;
run;
```

We recommend creating a special process for identification and cleaning of duplicate records in study data. It may start by running out-of-box data quality tools with validation results used as input to customized macro or *ad hoc* SAS programs to get more precise and detailed output. Review of validation reports may lead to either

- Adjust Key Variables → re-run SAS diagnostics programs, fix documentation (define.xml)
- Run data collection queries → contact Data Management team for data correction
- Fix SDTM mapping → re-run data mapping programs

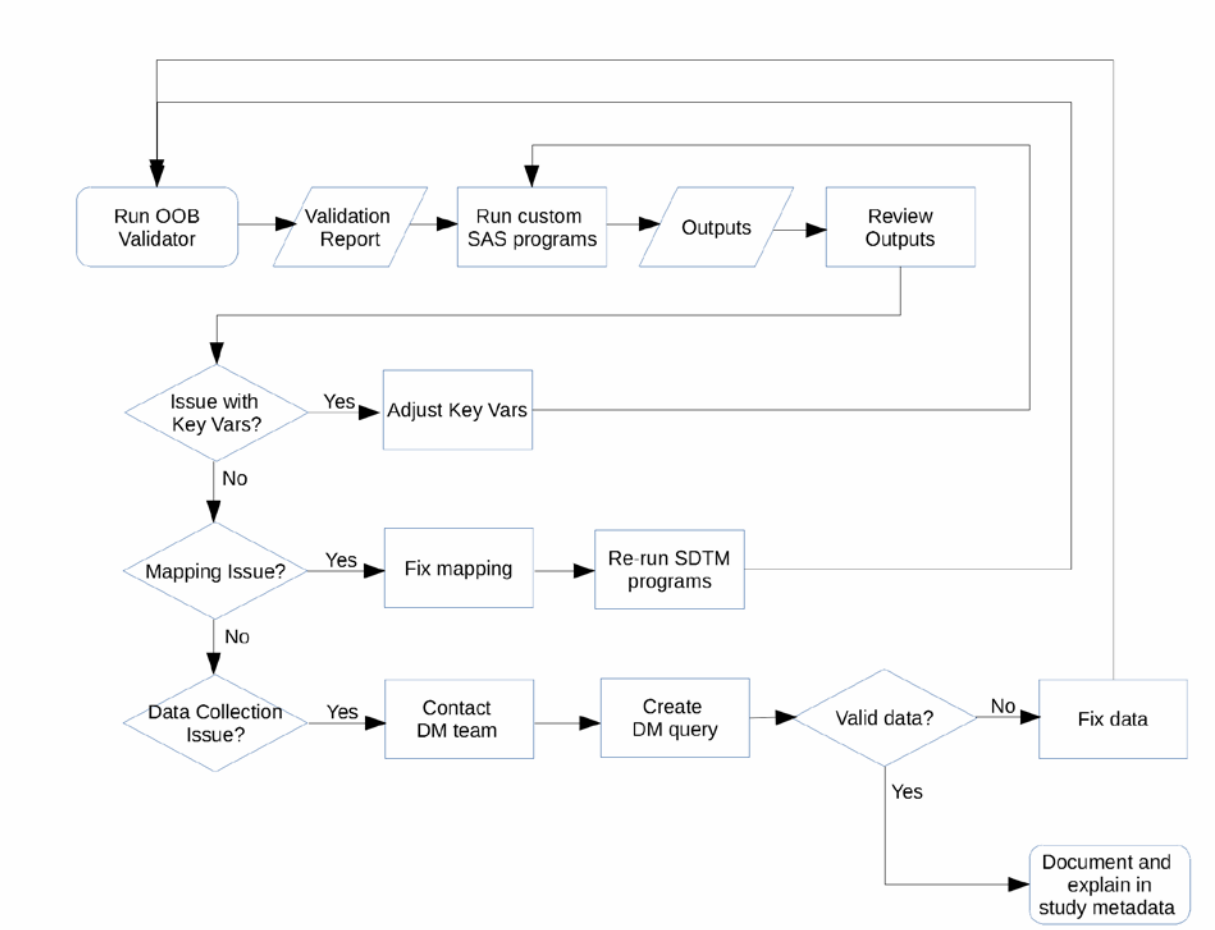


**Figure 1. Process for cleaning duplicate records**

There are many challenges in handling duplicate record issues. Usually there are multiple sources for duplicate records even within one dataset. We will discuss the most common examples in the next section. New data transfers may introduce new reasons for presence of duplicate records. If database is locked, it's too late to fix data collection errors. Therefore, data cleaning should start as early as possible. It's too difficult for Data Management team to catch duplicate records issues and they need help from Programmers.

## TYPICAL EXAMPLES

Standard datasets have too many variables to be all included into examples provided in this paper. Therefore we dropped some variables where records have the same values. Only important info is included for illustrational purposes.

The most common case is when data for Findings domain includes both actual results and *NOT DONE* records for the same assessment time point.

| USUBJID | LBTESTCD | LBORRES | LBORRESU | LBSTAT | LBNAM | VISIT | LBDTC |
|---|---|---|---|---|---|---|---|
| 001-001 | ALB | 4.7 | g/dL | | ABC | UNSCHEDULED | 2008-12-15 |
| 001-001 | ALB | | | NOT DONE | ABC | UNSCHEDULED | 2008-12-15 |
| 001-001 | ALP | 97 | IU/L | | ABC | UNSCHEDULED | 2008-12-15 |
| 001-001 | ALP | | | NOT DONE | ABC | UNSCHEDULED | 2008-12-15 |

**Table 1. Example of duplicates due to *NOT DONE* records and actual results**

This looks like a data management issue due to errors in data transfer from the lab. A programmer should contact the data management team for issue resolution before database is locked.

Sometimes duplicates are reported for multiple *NOT DONE* records with missing Timing info.

| USUBJID | VSTESTCD | VSORRES | VSORRESU | VSSTAT | VISIT | VSDTC |
|---|---|---|---|---|---|---|
| 001-001 | SYSPB | | | NOT DONE | | |
| 001-001 | SYSPB | | | NOT DONE | | |
| 001-001 | SYSPB | | | NOT DONE | | |
| 001-001 | SYSPB | | | NOT DONE | | |

**Table 2. Example of duplicates due to missing Timing info for *NOT DONE* records**

This could either be a data collection or SDTM mapping/programming error. If Visit Number is collected on CRF, then in cases of missing visits all empty Case Report Forms may be converted into *NOT DONE* records without specifying references to particular time point. To avoid this problem a proper data collection design (CRF) and implementation (EDC) should be applied to ensure that Timing info is populated for any assessments including *NOT DONE* records. At least study visit info (VISITNUM) is expected in such cases. Note, that submission data should include only collected information. Non-completed CRFs cannot be deemed as *collected data*. Consider if inclusion of this information is actually needed.

Another common source for duplicate records is due to different visits on the same time point.

| USUBJID | LBTESTCD | LBORRES | LBORRESU | LBNAM | VISIT | LBDTC |
|---|---|---|---|---|---|---|
| 001-001 | ALB | 4.7 | g/dL | ABC | Visit 3 | 2016-05-13T10:27 |
| 001-001 | ALB | 4.4 | g/dL | ABC | Unscheduled 3.1 | 2016-05-13T10:27 |
| 001-001 | ALP | 97 | IU/L | ABC | Visit 3 | 2016-05-13T10:27 |
| 001-001 | ALP | 101 | IU/L | ABC | Unscheduled 3.1 | 2016-05-13T10:27 |

**Table 3. Example of duplicates due to different Visit on the same time point**

Note, that in this example, Lab Sample Collection time point is provided with accuracy to minutes. How is it possible for subject to have two different visits within one minute? Such data structure introduces confusion in data interpretation and raises questions about overall compliance with study protocol. Data management team should explore this data issue and try to resolve it if possible. If not, then they need to create a detailed explanation for this data issue and its impact on analysis in the Reviewer's Guide.

The most confusing case of duplicate records is a presence of significantly different results on the same assessment time point.

| USUBJID | LBTESTCD | LBORRES | LBORRESU | LBNRIND | LBFAST | VISIT | LBDTC |
|---|---|---|---|---|---|---|---|
| 001-001 | GLUC | 96 | mg/dL | NORMAL | N | Unscheduled | 2012-03-02T22:40 |
| 001-001 | GLUC | 181 | mg/dL | HIGH | N | Unscheduled | 2012-03-02T22:40 |
| 001-001 | GLUC | 131 | mg/dL | HIGH | N | Unscheduled | 2012-03-02T22:40 |
| 001-001 | GLUC | 209 | mg/dL | HIGH | N | Unscheduled | 2012-03-02T22:40 |
| 001-001 | GLUC | 67 | mg/dL | NORMAL | N | Unscheduled | 2012-03-02T22:40 |
| 001-001 | GLUC | 81 | mg/dL | NORMAL | N | Unscheduled | 2012-03-02T22:40 |

**Table 4. Example of duplicates due to different results on the same time point**

In this example, a subject has 6 records for unscheduled assessment for blood *Glucose.* According to submitted data all 6 samples were collected within one-minute period. However, the results differ by almost 3.5 times. Three assessments are *NORMAL*, but another three are *ABNORMAL*. Such severe problems may raise a red flag for reviewers if they can trust the submission data?

We observed submissions with duplicate records in lab data with different Fasting Status info

| USUBJID | LBTESTCD | LBORRES | LBORNRLO | LBORNRHI | LBNRIND | LBFAST | LBDTC |
|---|---|---|---|---|---|---|---|
| 001-001 | GLUC | 3.9 | 4.6 | 6.4 | L | Y | 2011-04-13T10:45 |
| 001-001 | GLUC | 3.9 | 3.6 | 7.7 | | N | 2011-04-13T10:45 |

**Table 5. Example of duplicates due to different Fasting Status on the same time point**

Note that in this example, both records have the same collected results. However, Normal Ranges for blood *Glucose* assessments are different based on subject Fasting Status. Therefore, the first record is qualified as an abnormally low result, while another record is normal. In this example, a source for presence of duplicate records is still unclear and needs additional investigation by data management team. At the same time, this output reveals additional data management problems with accurate collection of Fasting Status info during the study.

In some cases, reported duplicate records have different standard units for test results.

| USUBJID | LBTESTCD | LBTEST | LBSTRESN | LBSTRESU | LBDTC |
|---|---|---|---|---|---|
| 001-001 | EOSLE | Eosinophils/Leukocytes | 0.2 | K/cu mm | 2012-07-08 |
| 001-001 | EOSLE | Eosinophils/Leukocytes | 2.5 | % | 2012-07-08 |

**Table 6. Example of duplicates with different standard units on results**

This is another example when diagnostics for duplicate records reveal issues with data collection and SDTM mapping. Lab results must have the same standard units for the same test. Note that a definition of "*Test*" is not limited to LBTESTCD variable and may include Method (LBMETHOD), Specimen (LBSPEC) or LOINC Code (LBLOINC) info. However, here is a clear invalid usage of *Eosinophils/Leukocytes* test name for different *Eosinophils* (Eosinophil Count) test. A source for this issue may be due to errors in data transfer from the lab. In general, physicians are OK with interpreting lab test results where an exact definition for test is represented by result units. However, such approach of using incorrect test names is absolutely unacceptable in preparation of standardized data. A programmer should contact Data Management team for fixing this data collection issue. If database is already locked, then error must be fixed in SDTM mapping. Detailed mapping documentation is extremely important in such cases to ensure compliance with CFR 21 Part 11 regulations.

There is also another case of duplicate records with different original units

| USUBJID | LBTEST | LBORRES | LBORRESU | LBSTRESN | LBSTRESU | LBDTC |
|---|---|---|---|---|---|---|
| 001-001 | Hemoglobin | 15 | g/dL | 150 | g/L | 2012-03-06T10:10 |
| 001-001 | Hemoglobin | 15 | mmol/L | 241.5 | g/L | 2012-03-06T10:10 |

**Table 7. Example of duplicates with different original units on the same results**

In this example, reported values for test are the same, but original result units are different. A conversion into standard unit leads to significantly different results, which will be used for analysis. It looks like a data collection error and must be fixed by Data Management team.

Exact duplicate records are still a common case

| USUBJID | LBSEQ | LBTEST | LBORRES | LBORRESU | LBDTC |
|---|---|---|---|---|---|
| 001-001 | 1 | Hemoglobin | 15 | g/dL | 2012-03-06T10:10 |
| 001-001 | 2 | Hemoglobin | 15 | g/dL | 2012-03-06T10:10 |

**Table 8. Example of exact duplicate records**

The only difference in this example is values in LBSEQ variable, which cannot be considered as a natural key in LB domain. Investigation for the source of such exact duplicates and a fix by Data Management team is expected. If database is already locked, the only potential solution may be data cleaning of exact duplicates during SDTM mapping. However, a company should evaluate risk/benefits of this data manipulation and follow CFR 21 Part 11 regulations.

In some cases programmers try to fix duplicate records artificially

| USUBJID | LBREFID | LBTEST | LBORRES | LBORRESU | LBDTC |
|---|---|---|---|---|---|
| 001-001 | 1 | Hemoglobin | 15 | g/dL | 2012-03-06T10:10 |
| 001-001 | 2 | Hemoglobin | 15 | g/dL | 2012-03-06T10:10 |

**Table 9. Example of exact duplicate records**

In this example, a sponsor provided an explanation in Reviewer's Guide: "*Adding LBREFID creates a unique sort*". However, there is no difference between LBREFID variable and LBSEQ variable in previous example. Both variables represent surrogate keys. Introducing LBREFID variable does not explain why duplicate records exist?

There are valid cases for usage of --REFID variable as a natural key. For example, LBREFID may represent Lab Specimen Tracking ID. While multiple results on the same time point still introduce complexity in their interpretation, clear definition of data structure helps reviewers.

Here is an example when different labs are used in a study

| USUBJID | LBREFID | LBTEST | LBORRES | LBORRESU | LBNAM | LBDTC |
|---|---|---|---|---|---|---|
| 001-001 | 12345 | Hemoglobin | 18 | g/dL | Central Lab | 2012-03-06T10:10 |
| 001-001 | 12346 | Hemoglobin | 15 | g/dL | Local Lab | 2012-03-06T10:10 |

**Table 10. Example of results from different labs**

This is a completely valid case and correct implementation. Test definition may include a reference to lab-performed assessment of collected samples. We use this example to emphasize a need for good documentation. LBNAM should be included in a list of key variables for LB domain in define.xml file. Additional information on how lab test results data were analyzed in relation to different labs should be included in Reviewer's Guide. For example, what lab was used in reporting? Safety? Efficacy? Patient Screening? What if results from different labs were both Normal and Abnormal?

Most provided examples are based on Findings data. However some cases are also applicable for Events and Interventions. For example, missing Timing info, exact duplicates, etc. Events data has a special case of duplicate records with the same collected Terms and Start Dates

| USUBJID | AETERM | AESEV | AESTDTC | AEENDTC | VISITNUM |
|---|---|---|---|---|---|
| 001-001 | Headache | Mild | 2012-01-01 | | 1 |
| 001-001 | Headache | Mild | 2012-01-01 | | 2 |
| 001-001 | Headache | Severe | 2012-01-01 | | 3 |
| 001-001 | Headache | Mild | 2012-01-01 | | 4 |
| 001-001 | Headache | Moderate | 2012-01-01 | 2012-06-01 | 5 |

**Table 11. Example of a single Adverse Events with assessments on each visit**

In this example, sponsor collected info about ongoing Adverse Events as continuous assessments during each subject visit. While it's a valid data collection process, a mapping to SDTM data was incorrect. SDTM Implementation Guide has examples on how to handle such information in SDTM structure. The first option would to create one AE record with maximum Severity observed and provide additional details on AE in Finding About (FA) domain. Another option would be to merge records with the same AE attributes, like Severity, for particular interval into one record. For example, records #1 and #2. However each AE record should have unique Start Date, which is expected to be equal to End Date of the previous record for a continuous Adverse Event.

## CONCLUSION

Duplicate records are a major issue effecting regulatory review and analysis. Identifying duplicate record issues as soon as possible and fixing them before data lock is a best practice. We recommend establishing a process for detecting and resolving duplicate record issues that is a standard part of the Data Management Plan for each study.

An optimal process must ensure the correctness of study metadata, so that out-of-the-box validation tools can be used to detect duplicate records. These tools will only works as good as the quality of your metadata.

Collaboration between Data Management and SAS Programmers is also a critical factor in establishing the process that is able to detect duplicate records early and fix them before data lock. In many cases, duplicate records are difficult to catch using standard data management tools, thus SAS Programmers must be involved to assist in the process of data cleaning.

Finally, documenting Key Variables in define.xml, including detailed variable descriptions, and providing study-specific details in Reviewer's Guide is important in assisting regulatory reviewers in interpretation of study data.

## REFERENCES

[1] Elmagarmid, Ahmed K. 2007. *Duplicate Record Detection: A Survey*. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 19, NO. 1. Available at http://www.ipeirotis.com/wp-content/uploads/2012/01/tkde2007.pdf

[2] Doi, Mary. "*How Good is Your SDTM Data? Perspectives from JumpStart*". PhUSE CSS. March 2016. Available at http://www.phusewiki.org/docs/CSS%202016%20Presentations/SDTM%20Mary%20Doi.pptx

[3] CDISC SDTM Implementation Guide. Available at http://www.cdisc.org/sdtm

[4] Pinnacle 21 Community. Available at www.opencdisc.org

[5] Pinnacle 21 Enterprise. Available at www.pinnacle21.net

## CONTACT INFORMATION

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

Name: Sergiy Sirichenko
Company: Pinnacle 21 LLC
Work Phone: 908-781-2342
E-mail: ssirichenko@pinnacle21.net

Name: Max Kanevsky
Company: Pinnacle 21 LLC
Work Phone: 267-331-4431
E-mail: mkanevsky@pinnacle21.net