



Constrain your Terminology

Use Integrity Constraints to Validate your Data

Sven Greiner, HMS Analytical Software GmbH

PhUSE 2017 Edinburgh – Paper DH06





SDTM and ADaM require the use of Controlled Terminology (CT)

How do you ensure the consistency of data and CT?

- Software (e.g. Pinnacle21)
- Define.xml macro
- Integrity Constraints (ICs)



Integrity Constraints



“Integrity constraints are a set of data validation rules that ... restrict the data values ... in a SAS data file.” [1]

[1] SAS Language Reference: Concepts, Second Edition, Understanding Integrity Constraints

General Integrity Constraints

General ICs are data validation rules for a single SAS dataset

There are four types of general ICs:

- CHECK:** Limit variable values to a list or range of values
- NOT NULL:** Missing values are not allowed in a variable
- UNIQUE:** A unique combination of values is required in the specified variable(s)
- PRIMARY KEY:** Unique combination of non-missing values in the specified variable(s)

CHECK Example

An example of a CHECK IC added to dataset ADSL for variable SEX:

```
proc datasets;  
  modify adsl;  
  ic create con_sex =  
    check(where=(sex in ('M', 'F')))  
    message = "Valid values for SEX are either 'M' or 'F'. ";  
quit;
```

IC name

Variable name

List of valid values

User-defined error message

ADS Specifications Example

Key	Variable Name	Variable Label	Variable Type	Variable Length	Variable Format	Controlled Terminology
1	USUBJID	Unique Subject Identifier	text	20	\$20.	
	AGE	Age	num	8		
	SEX	Sex	text	1	\$1.	SEXC
	RACE	Race	text	20	\$20.	RACEC
	TRTP	Planned Treatment	text	20	\$20.	TRTPC

“M” = “Male”
“F” = “Female”

“White” = “White”
“Indian” = “Indian”
“Black” = “Black”

“TRT_A” = “TRT_A”
“TRT_B” = “TRT_B”
“ ” = “ ”

Integrity Constraints in Action

Step 1 - Create Template Dataset

```
proc sql;
  create table adsl(label='Subject-Level Analysis Dataset')
  {
    usubjid char (20) label="Unique Subject Identifier" format=$20.,
    age      num  (8) label="Age",
    sex      char (1) label="Sex" format=$1.,
    race     char (20) label="Race" format=$20.,
    trtp     char (20) label="Planned Treatment" format=$20.
  }
  constraint CON_SEX check(sex in ('M','F'))
    message="Valid values for variable SEX: 'M', 'F'",
  constraint CON_RACE check(race in ('White','Indian','Black'))
    message = "Valid values for variable RACE: 'White', 'Indian', 'Black'",
  constraint CON_TRTP check(trtp in ('TRT_A','TRT_B',''))
    message = "Valid values for variable TRTP: 'TRT_A', 'TRT_B', ''",
  constraint UNIQUE_KEYS unique(usubjid));
quit;
```

Integrity Constraints in Action

Step 2 - Initiate Audit Trail

```
proc datasets library=work nolist;  
    audit adsl;  
    initiate;  
quit;
```

The audit trail

- logs modifications to a SAS dataset, and
- stores observations that were rejected by ICs.

Integrity Constraints in Action

Step 3 - Create Derived Data

```
data dummy;  
  length usubjid $20 age 8 sex $1 race $20 trtp $20;  
  input usubjid age sex race trtp;  
  datalines;  
ABC-001 55 M White TRT_A  
ABC-002 26 F Blue TRT_B  
ABC-003 45 M Indian TRT_C;  
run;
```

- Derived data to be inserted into the ADSL template dataset
- The red strings indicate an IC violation

Integrity Constraints in Action

Step 4.1 - Insert Data into the Template Dataset

```
proc sql;  
  insert into adsl  
  select * from dummy;  
quit;
```

ABC-001	55	M	White	TRT_A
ABC-002	26	F	Blue	TRT_B
ABC-003	45	M	Indian	TRT_C

ERROR: Valid values for variable RACE: 'White', 'Indian', 'Black'

NOTE: Deleting the successful inserts before error noted above to restore table to a consistent state.

NOTE: The SAS System stopped processing this step because of errors.

Integrity Constraints in Action

Step 4.2 - Count Observations

```
proc sql noprint;  
    select count(*)  
    into :nobs  
    from dummy;  
quit;
```

Count number of observations to add into ADSL

Integrity Constraints in Action

Step 4.3 - Insert Data into the Template Dataset

```
%macro insert_obs(nobs=);  
proc sql;  
%do i = 1 %to &nobs.;  
    insert into adsl  
    select * from dummy(firstobs=&i. obs=&i.);  
%end;  
quit;  
%mend insert_obs;  
  
%insert_obs(nobs=&nobs.);
```

ABC-001	55	M	White	TRT_A
ABC-002	26	F	Blue	TRT_B
ABC-003	45	M	Indian	TRT_C

NOTE: 1 row was inserted into WORK.ADSL.

ERROR: Valid values for variable RACE: 'White', 'Indian', 'Black'

NOTE: Deleting the successful inserts before error noted above to restore table to a consistent state.

ERROR: Valid values for variable TRTP: 'TRT_A', 'TRT_B', "

NOTE: Deleting the successful inserts before error noted above to restore table to a consistent state.

NOTE: The SAS System stopped processing this step because of errors.

Integrity Constraints in Action

Step 5 - Print Audit Trail

```
data audit(drop=_atdatetime_ _atobsno_ _atreturncode_  
           _atuserid_ _atopcode_);  
  set adsl(type=audit);  
  where _atopcode_ eq "EA";  
run;  
  
proc print data=audit noobs;  
  id _atmessage_;  
run;
```

Observation add failed

The SAS System

ATMESSAGE	usubjid	age	sex	race	trtp
ERROR: Valid values for variable RACE: 'White', 'Indian', 'Black'	ABC-002	26	F	Blue	TRT_B
ERROR: Valid values for variable TRTP: 'TRT_A', 'TRT_B', "	ABC-003	45	M	Indian	TRT_C

Integrity Constraints in Action

Step 6 - Create complete ADS

```
proc datasets nolist;  
    modify adsl;  
→ ic delete _all_;  
quit;
```

```
proc sql;  
→ delete from adsl;  
→ insert into adsl select * from dummy;  
quit;
```

- Remove ICs and observations from the template dataset, and
- insert data without IC checks.

Integrity Constraints in Action

Summary

- ICs can check datasets against CT
- Inserting observations
 - all at once results in only the first violation being reported
 - individually results in performance issues for larger datasets
- Multiple violations in one observation cannot be identified
- Log or audit trail track violations (max. 180 characters)
- Complete re-creation of the dataset is advised after IC checks



Conclusion



ICs

- Enable you to check your data against the CTs while developing your ADSs
- Have several technical limitations

Alternatives like define.xml macros or Pinnacle21

- Are not run during ADS creation, but afterwards
- Results in inconsistencies between ADS and CT





A suitable solution

- A macro that checks a variable against the list of valid values
 - E.g. If sex not in (,M',F') then put ...;
- Promises better performance and identification of CT-deviations than ICs

Thank you very much for your attention!



Sven Greiner

Senior Statistical Programmer

Sven.Greiner@analytical-software.de



HMS Analytical Software GmbH
Rohrbacher Str. 26
69115 Heidelberg
www.analytical-software.de
info@analytical-software.de
+49-6221-6051-0