

m_val_chk_duplicates.sas File Reference

Validation

Validation macro to check data by using a duplicate values rule

Description

The macro can be used to validate data by using a rule to detect duplicate values of columns in a given SAS dataset or database table. The COL_LIST parameter may contain one or more columns from the source table. The routine detects the existence of the columns in the given source table automatically. The columns in the list are stored including its value in the exception table. If the output target, error or exception tables are not defined by their according parameter values, output tables are defined by the macro routine and created in the sessions WORK library. If the exception table is given, the exceptions found by the validation will be appended to the given exception table per default. The same procedure is used for the error table, just as long as the ACTION parameter value is not equal to "Check".

Note

If the PRINT parameter value is set to Y, a SAS proc report step is used to print the validation summary status on the result tab of SAS Enterprise Guide or Stored Process Server.

Autors

Paul Alexander Canals y Trocha (paul.canals@gmail.com)

Date

2021-03-27 00:00:00

Version

21.1.03

Link

<https://github.com/paul-canals/toolbox>

Parameters

Input	help	Parameter, if set (Help or ?) to print the Help information in the log. In all other cases this parameter should be left out from the macro call.
Input	src_tbl	Specifies the LIBNAME.TABLENAME of the source SAS dataset or database table on which the validation is performed. The default value is: <code>_NONE_</code> .
Input	tgt_tbl	Specifies the LIBNAME.TABLENAME of the target SAS dataset or database table in which the valid data is stored. If this target table is not provided, this target table is created automatically in the WORK library. The default value is: <code>_NONE_</code> .
Input	err_tbl	Specifies the LIBNAME.TABLENAME of the target SAS dataset or database table in which the error data is stored. If this target table is not provided, this target table is created automatically in the WORK library. The default value is: <code>_NONE_</code> .
Input	exc_tbl	Specifies the LIBNAME.TABLENAME of the target SAS dataset or database table in which the exceptions are stored. If this target table is not provided, this target table is created automatically in the WORK library. The default value is: <code>_NONE_</code> .
Input	col_list	Specifies the list of columns separated by a blank character to validate for duplicate column values.
Input	action	Indicator [CHECK MOVE ABORT] to specify the to be taken by the validation routine. When the ACTION value is set to "Check", the source table records are validated and found exceptions written to the exception table, but also all records will be written to the output target table. When set to "Move", the error records are written to the error table and only valid records to the target table. When set to "Abort" only the found exceptions are written to the exception table, but not into any other target tables. The default value is: Check.
Input	append	Boolean [Y N] parameter value to specify if the found exceptions (and errors) are to be appended into a given exception table (and error table). The default value for APPEND is: Y.
Input	print	Boolean [Y N] parameter to generate the output by a SAS proc report step with style HtmlBlue. The default value for PRINT is: N.
Input	debug	Boolean [Y N] parameter to provide verbose mode information. The default value is: N.

Returns

- Validation result target tables.

Calls

- [m_utl_chg_delimiter.sas](#)
- [m_utl_chk_reqcols.sas](#)
- [m_utl_chk_table_exist.sas](#)
- [m_utl_get_col_code.sas](#)
- [m_utl_nlobs.sas](#)
- [m_utl_print_message.sas](#)
- [m_utl_print_mtrace.sas](#)
- [m_utl_varlist.sas](#)

Usage

Example 1: Show help information:

```
%m_val_chk_duplicates(?)
```

Example 2: Validate SASHELP.class to check if all student names are unique:

```
%m_val_chk_duplicates(  
  src_tbl = SASHELP.class  
  , col_list = Name  
  , action = Check  
  , print = Y  
  , debug = Y  
);
```

Example 3: Validate SASHELP.class to ensure all student info is unique:

```
%m_val_chk_duplicates(  
  src_tbl = SASHELP.class  
  , col_list = Weight Height  
  , action = Move  
  , print = Y  
  , debug = N  
);  
  
proc print data=WORK.class_exc label;  
run;
```

Example 4: Validate SASHELP.class to check if all student info is unique:

```
%m_val_chk_duplicates(  
  src_tbl = SASHELP.class  
  , exc_tbl = WORK.class_dups  
  , col_list = Weight  
  , action = Check  
  , print = Y  
  , debug = N  
);  
  
%m_val_chk_duplicates(  
  src_tbl = SASHELP.class  
  , exc_tbl = WORK.class_dups  
  , col_list = Height  
  , action = Check  
  , print = Y  
  , debug = N  
);  
  
proc print data=WORK.class_dups label;  
run;
```

Copyright

Copyright 2008-2021 Paul Alexander Canals y Trocha.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.