Paper 261-30

Manipulating Data with PROC SQL

Kirk Paul Lafler, Software Intelligence Corporation

ABSTRACT

PROC SQL is a popular database language with numerous extensions for working with numeric and character data including an assortment of operators, functions, and predicates. This paper presents coding techniques that perform text case conversions, concatenation of two or more character strings, pattern and phonetic matching operations, updates to data in a table, and other useful coding techniques for SAS and PROC SQL users.

INTRODUCTION

PROC SQL provides numerous arithmetic, statistical, and summary functions to manipulate numeric data. With one numeric data type to represent numeric data, the NUMERIC or NUM column definition is automatically assigned a default length of 8 bytes, even if the column is created with a numeric length less than 8 bytes.

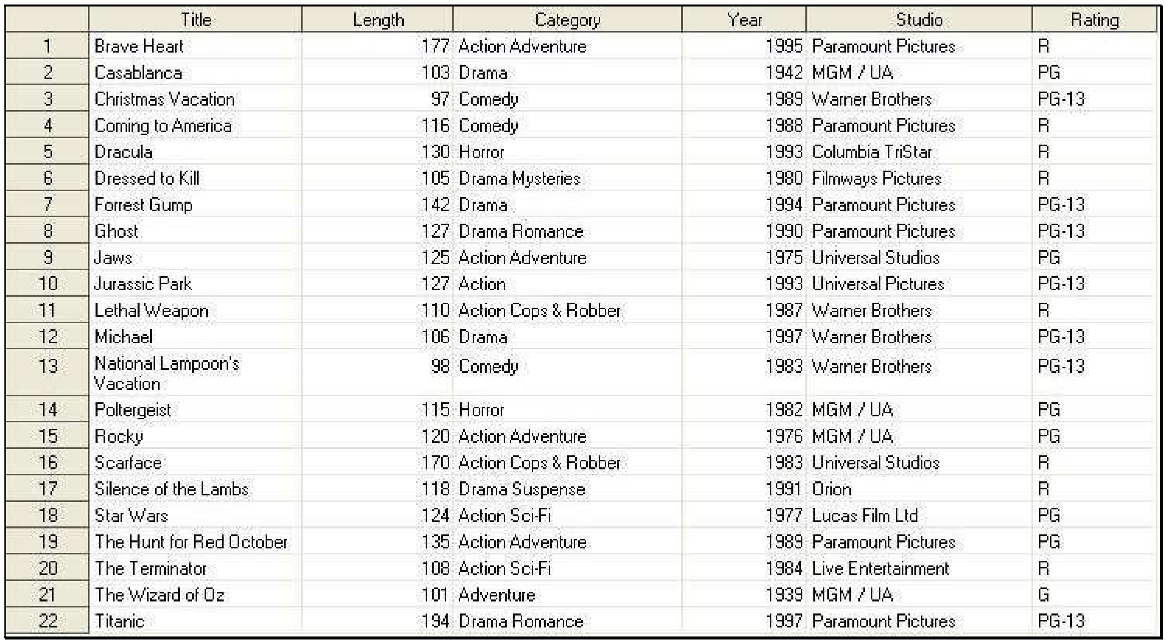
PROC SQL also provides numerous functions and operators to manipulate character data including words, text, and codes using the CHARACTER or CHAR data type. The CHARACTER or CHAR data type allows ASCII or EBCDIC character sets and stores fixed-length character strings consisting of a maximum of 32K characters.

The PROC SQL user has a vast array of functions and operators that can make the task of working with numeric and character data considerably easier. This paper will illustrate how columns based on the numeric and character data types are defined; how string functions, pattern matching, phonetic matching techniques, and a variety of other techniques are used with numeric and character data.

DATA USED IN EXAMPLES

The data used in all the examples in this paper consists of a selection of movies that I've viewed over the years. The Movies table consists of six columns: title, length, category, year, studio, and rating. Title, category, studio, and rating are defined as character columns with length and year being defined as numeric columns. The data stored in the Movies table is depicted below.

MOVIES Table



	Title	Length	Category	Year	Studio	Rating
1	Brave Heart	177	Action Adventure	1995	Paramount Pictures	R
2	Casablanca	103	Drama	1942	MGM / UA	PG
3	Christmas Vacation	97	Comedy	1989	Warner Brothers	PG-13
4	Coming to America	116	Comedy	1988	Paramount Pictures	R
5	Dracula	130	Horror	1993	Columbia TriStar	R
6	Dressed to Kill	105	Drama Mysteries	1980	Filmways Pictures	R
7	Forrest Gump	142	Drama	1994	Paramount Pictures	PG-13
8	Ghost	127	Drama Romance	1990	Paramount Pictures	PG-13
9	Jaws	125	Action Adventure	1975	Universal Studios	PG
10	Jurassic Park	127	Action	1993	Universal Pictures	PG-13
11	Lethal Weapon	110	Action Cops & Robber	1987	Warner Brothers	R
12	Michael	106	Drama	1997	Warner Brothers	PG-13
13	National Lampoon's Vacation	98	Comedy	1983	Warner Brothers	PG-13
14	Poltergeist	115	Horror	1982	MGM / UA	PG
15	Rocky	120	Action Adventure	1976	MGM / UA	PG
16	Scarface	170	Action Cops & Robber	1983	Universal Studios	R
17	Silence of the Lambs	118	Drama Suspense	1991	Orion	R
18	Star Wars	124	Action Sci-Fi	1977	Lucas Film Ltd	PG
19	The Hunt for Red October	135	Action Adventure	1989	Paramount Pictures	PG
20	The Terminator	108	Action Sci-Fi	1984	Live Entertainment	R
21	The Wizard of Oz	101	Adventure	1939	MGM / UA	G
22	Titanic	194	Drama Romance	1997	Paramount Pictures	PG-13

SQL OPERATORS AND FUNCTIONS

PROC SQL users have a number of ways to accomplish their objectives, particularly when the goal is to manipulate data. The SELECT statement is an extremely powerful statement in the SQL language. Its syntax can be somewhat complex because of the number of ways that columns, tables, operators, and functions can be combined into executable statements. There are several types of operators and functions in PROC SQL: 1) comparison operators, 2) logical operators, 3) arithmetic operators, 4) character string operators, and 5) summary functions.

Comparison Operators

Comparison operators are used in the SQL procedure to compare one character or numeric values to another. As in the DATA step, PROC SQL comparison operators, mnemonics, and their descriptions appear in the following table.

SAS Operator	Mnemonic Operators	Description
=	EQ	Equal to
^= or ¬=	NE	Not equal to
<	LT	Less than
<=	LE	Less than or equal to
>	GT	Greater than
>=	GE	Greater than or equal to

Say, for example, that you wanted to select only those movies from the MOVIES table that had a running length longer than 2 hours (120 minutes). The following example illustrates the use of the greater than sign (>) in a WHERE clause to select movies that meets the WHERE clause condition.

PROC SQL Code

```
PROC SQL;
  SELECT title, rating, length
    FROM wuss.movies
      WHERE length > 120;
QUIT;
```

Results

```
Title                            Rating    Length
_________________________________________________

Brave Heart                      R            177
Dracula                          R            130
Forrest Gump                     PG-13        142
Ghost                            PG-13        127
Jaws                             PG           125
Jurassic Park                    PG-13        127
Scarface                         R            170
Star Wars                        PG           124
The Hunt for Red October         PG           135
Titanic                          PG-13        194
```

Logical Operators

Another type of operator – known as a logical operator is used to connect two or more expressions together in a WHERE or HAVING clause. The available logical operators include AND, OR, and NOT. Say, for example, you wanted to select only those movies with a running length of more than 2 hours (>120 minutes) and a rating of “PG”. The next example illustrates how the AND operator is used to ensure that both conditions are true.

PROC SQL Code

```
PROC SQL;
  SELECT title, rating, length
    FROM wuss.movies
      WHERE length > 120 AND
            rating = “PG”;
QUIT;
```

Results

```
Title                           Rating    Length
------------------------------------------------

Jaws                            PG           125
Star Wars                       PG           124
The Hunt for Red October        PG           135
```

The next example illustrates the use of the OR logical operator to select movies with a running length of more than 120 minutes or a rating of “PG”.

PROC SQL Code

```
PROC SQL;
  SELECT title, rating, length
    FROM wuss.movies
      WHERE length > 120 OR
            rating = “PG”;
QUIT;
```

Results

```
Title                           Rating    Length
------------------------------------------------

Brave Heart                     R            177
Casablanca                      PG           103
Dracula                         R            130
Forrest Gump                    PG-13        142
Ghost                           PG-13        127
Jaws                            PG           125
Jurassic Park                   PG-13        127
Poltergeist                     PG           115
Rocky                           PG           120
Scarface                        R            170
Star Wars                       PG           124
The Hunt for Red October        PG           135
Titanic                         PG-13        194
```

Arithmetic Operators

The arithmetic operators used in PROC SQL are the same as those used in the DATA step and other languages including C, Pascal, FORTRAN, and COBOL. The arithmetic operators available in the PROC SQL appear below.

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
**	Exponent (raises to a power)
=	Equals

To illustrate how arithmetic operators are used, suppose you desired to add ten minutes to the running length of each movie in the MOVIES table due to the splicing of corporate advertisements. The next example illustrates the use of the addition arithmetic operator with the definition of a user-defined column alias to accomplish this task.

PROC SQL Code

```
PROC SQL;
  SELECT title, rating, length, length + 20 AS Revised_Length
    FROM wuss.movies;
QUIT;
```

Results

```
                                                        Revised_
Title                           Rating    Length          Length
________________________________________________________________

Brave Heart                     R            177             197
Casablanca                      PG           103             123
Christmas Vacation              PG-13         97             117
Coming to America               R            116             136
Dracula                         R            130             150
Dressed to Kill                 R            105             125
Forrest Gump                    PG-13        142             162
Ghost                           PG-13        127             147
Jaws                            PG           125             145
Jurassic Park                   PG-13        127             147
Lethal Weapon                   R            110             130
Michael                         PG-13        106             126
National Lampoon's Vacation     PG-13         98             118
Poltergeist                     PG           115             135
Rocky                           PG           120             140
Scarface                        R            170             190
Silence of the Lambs            R            118             138
Star Wars                       PG           124             144
The Hunt for Red October        PG           135             155
The Terminator                  R            108             128
The Wizard of Oz                G            101             121
Titanic                         PG-13        194             214
```

Character String Operators and Functions

Character string operators and functions are typically used with character data. Numerous operators are presented to acquaint users to the power available with the SQL procedure. You'll see a number of operators including string concatenation, character alignment, .

Concatenation and Character Alignment

The default alignment for character data is to the left, however character columns or expressions can also be aligned to the right. Two functions are available for character alignment: LEFT and RIGHT. The next example combines the concatenation operator "||" and the TRIM function with the **LEFT** function to left align a character expression while inserting blank spaces and a dash "-" between two character columns to subset "PG-rated" movies.

PROC SQL Code

```
PROC SQL;
  SELECT LEFT(TRIM(title) || " - " || category) AS Concatenation_Alignment
    FROM wuss.movies
      WHERE rating = "PG";
QUIT;
```

Results

```
Concatenation_Alignment
_____________________________________________________
Casablanca - Drama
Jaws - Action Adventure
Poltergeist - Horror
Rocky - Action Adventure
Star Wars - Action Sci-Fi
The Hunt for Red October - Action Adventure
```

Phonetic Matching (Sounds-Like Operator =)*

A technique for finding names that sound alike or have spelling variations is available in PROC SQL. Although not technically a function, the sounds-like operator "=*" searches and selects character data based on two expressions: the search value and the matched value. Anyone that has looked for a last name in a local telephone directory is quickly reminded of the possible phonetic variations. To illustrate how the sounds-like operator works, we will search on the movie title in the MOVIES table using the string "Rucky" for any and all phonetic variations related to the movie title "Rocky".

PROC SQL Code

```
PROC SQL;
  SELECT title, rating, category
    FROM wuss.movies
      WHERE title =* "Rucky";
QUIT;
```

Results

```
Title                          Rating  Category
____________________________________________________________
Rocky                          PG      Action Adventure
```

Finding Patterns in a String (Pattern Matching % and _)

Constructing specific search patterns in string expressions is a simple process with the LIKE predicate. The % acts as a wildcard character representing any number of characters, including any combination of upper or lower case characters. Combining the LIKE predicate with the % (percent sign) permits case-sensitive searches and is a popular technique used by savvy SQL programmers to find patterns in their data. The next example finds patterns in the movie category containing the uppercase character 'D' in the first position followed by any number of characters in the CATEGORY column.

PROC SQL Code

```
PROC SQL;
  SELECT title, rating, category
    FROM wuss.movies
      WHERE category LIKE 'D%';
QUIT;
```

Results

```
Title                           Rating  Category
____________________________________________________________
Casablanca                      PG      Drama
Forrest Gump                    PG-13   Drama
Michael                         PG-13   Drama
Dressed to Kill                 R       Drama Mysteries
Ghost                           PG-13   Drama Romance
Titanic                         PG-13   Drama Romance
Silence of the Lambs            R       Drama Suspense
```

Summarizing Data

PROC SQL is a wonderful tool for summarizing (or aggregating) data. It provides a number of useful summary (or aggregate) functions to help perform calculations, descriptive statistics, and other aggregating operations in a SELECT statement or HAVING clause. These functions are designed to summarize information and not display detail about data.

A number of summary functions are available including facilities to count non-missing values; determine the minimum and maximum values in specific columns; return the range of values; compute the mean, standard deviation, and variance of specific values; and other aggregating functions. In the following table an alphabetical listing of the available summary functions are displayed and, when multiple names for the same function are available, the ANSI-approved name appears first.

Summary Function	Description
AVG, MEAN	Average or mean of values
COUNT, FREQ, N	Aggregate number of non-missing values
CSS	Corrected sum of squares
CV	Coefficient of variation
MAX	Largest value
MIN	Smallest value
NMISS	Number of missing values
PRT	Probability of a greater absolute value of Student's t
RANGE	Difference between the largest and smallest values
STD	Standard deviation
STDERR	Standard error of the mean
SUM	Sum of values
SUMWGT	Sum of the weight variable values which is 1
T	Testing the hypothesis that the population mean is zero
USS	Uncorrected sum of squares
VAR	Variance

The next example uses the **COUNT** function with the **(*)** argument to produce a total number of rows, regardless if data is missing. The asterisk (*) is specified as the argument to count all rows in the PURCHASES table.

PROC SQL Code

```
PROC SQL;
  SELECT COUNT(*) AS Row_Count
    FROM wuss.movies;
QUIT;
```

Results

```
Row_Count
ƒƒƒƒƒƒƒƒƒ
       22
```

The next example illustrates the **MIN** summary function being specified to determine what the shortest movie is in the MOVIES table.

PROC SQL Code

```
PROC SQL;
  SELECT MIN(length) AS Shortest_Movie
    FROM wuss.movies;
QUIT;
```

Results

```
 Shortest_
     Movie
ƒƒƒƒƒƒƒƒƒƒƒƒ
        97
```

Summarizing data down rows
PROC SQL can be used to produce a single aggregate value by summarizing data down rows. The advantage of using a summary function in PROC SQL is that it generally computes the aggregate quicker than if a user-defined equation were constructed, and it reduces the amount of program testing. Suppose you wanted to know the average length "PG-rated" movie in the MOVIES table containing a variety of movie ratings. The next query computes the average movie length and produces a single aggregate value using the **AVG** function.

PROC SQL Code

```
PROC SQL;
  SELECT AVG(length) AS Average_Movie_Length
    FROM wuss.movies
      WHERE UPCASE(rating) = "PG";
QUIT;
```

Results

```
    Average_
Movie_Length
ƒƒƒƒƒƒƒƒƒƒƒƒ
    120.3333
```

CONCLUSION

PROC SQL is essentially a database language as opposed to a procedural or computational language. Although only two data types are available in the SAS System's implementation of SQL – numeric and character, numerous extensions including operators, functions, and other features are available to PROC SQL users for the purpose of manipulating data.

REFERENCES

Lafler, Kirk Paul (2004). *PROC SQL: Beyond the Basics Using SAS*, SAS Institute Inc., Cary, NC, USA.

Lafler, Kirk Paul (2003), "*Undocumented and Hard-to-find PROC SQL Features*," *Proceedings of the Eleventh Annual Western Users of SAS Software Conference*.

Lafler, Kirk Paul (1992-2004). *PROC SQL for Beginners*, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (1998-2004). *Intermediate PROC SQL*. Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2001-2004), *Advanced PROC SQL*. Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2002). *PROC SQL Programming Tips*. Software Intelligence Corporation, Spring Valley, CA, USA.

SAS® Guide to the SQL Procedure: Usage and Reference (1990). Version 6, First Edition; SAS Institute, Cary, NC, USA.

SAS® SQL Procedure User's Guide, Version 8 (2000). SAS Institute Inc., Cary, NC, USA.

ACKNOWLEDGMENTS

I would like to thank Lori Griffin and Andy Kuligowski (Tutorials Section Co-Chairs) for accepting my abstract and paper, as well as Greg Nelsen (Conference Chair), and the SUGI Leadership for their support of a great Conference.

CONTACT INFORMATION

Kirk Paul Lafler, a SAS Certified Professional® and former SAS Alliance Partner® (1996 - 2002) with 27 years of SAS software experience, provides consulting services and hands-on SAS training around the world. Kirk has written four books including PROC SQL: Beyond the Basics Using SAS (2004) by SAS Institute, Power SAS (2002) and Power AOL (2002) by Apress, and more than one hundred articles in professional journals and SAS User Group proceedings. His popular SAS Tips column appears regularly in the BASAS, HASUG, SANDS, SAS, SESUG, and WUSS Newsletters and websites. Kirk welcomes comments and can be reached at:

Kirk Paul Lafler
Software Intelligence Corporation
P.O. Box 1390
Spring Valley, California 91979-1390
Voice: 619-277-7350
E-mail: KirkLafler@cs.com
Web: www.software-intel.com