Paper RF-02-2015

Essential PROC SQL Join Techniques Using SAS® University Edition Software

Kirk Paul Lafler, Software Intelligence Corporation, Spring Valley, California

Abstract

After installing SAS Institute's free "SAS University Edition" you'll want to test drive the software. SAS University Edition includes Base SAS, SAS/STAT, SAS/IML, Designer Studio (user interface), and SAS/ACCESS for Windows, with all the powerful features found in the licensed SAS versions. To demonstrate the power found within SAS University Edition, we present conventional and unconventional PROC SQL join programming techniques using Base SAS software. All SAS users are encouraged to attend and learn essential concepts, syntax and programming techniques.

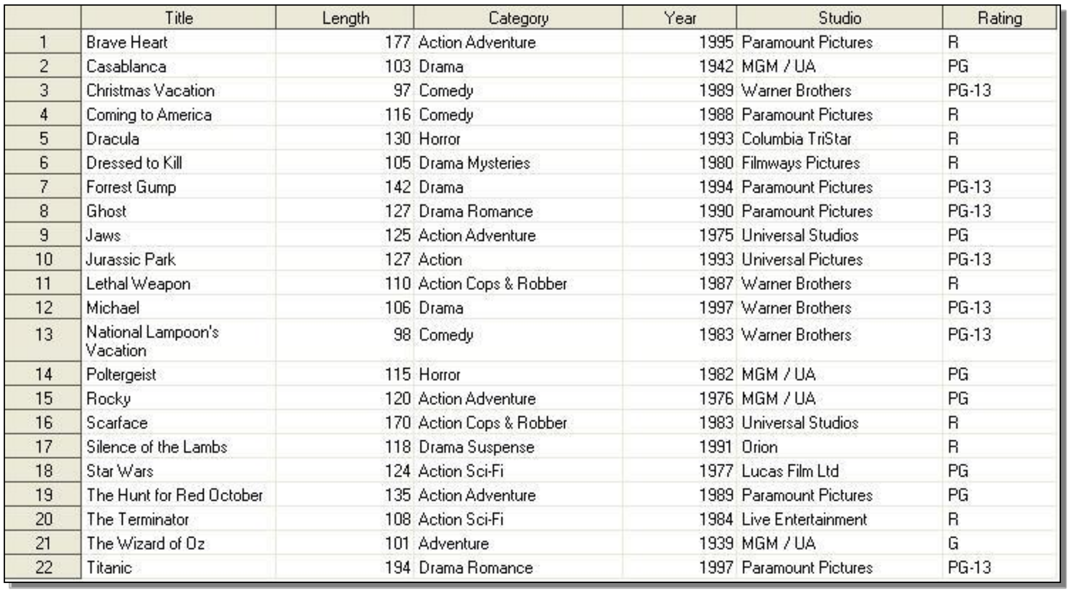
Introduction

A powerful and essential Base-SAS programming technique that all SAS users should be aware of, and comfortable performing, is the process of joining (or combining) two or more tables. The intent of this paper is to describe the join process, including what a join is, the preparation requirements for each table being specified in a join, the join syntax, and the various types of joins available to SAS users.

Example Tables

The examples used throughout this paper utilize a database of two tables. (A relational database is a collection of tables.) 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.

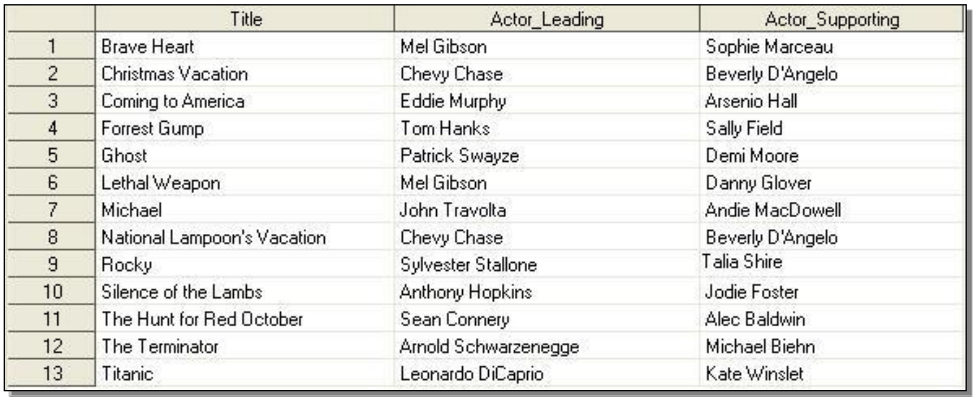
MOVIES Table



	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

The data stored in the ACTORS table consists of three columns: title, actor_leading, and actor_supporting, all of which are defined as character columns. The data stored in the Actors table is illustrated below.

ACTORS Table



	Title	Actor_Leading	Actor_Supporting
1	Brave Heart	Mel Gibson	Sophie Marceau
2	Christmas Vacation	Chevy Chase	Beverly D'Angelo
3	Coming to America	Eddie Murphy	Arsenio Hall
4	Forrest Gump	Tom Hanks	Sally Field
5	Ghost	Patrick Swayze	Demi Moore
6	Lethal Weapon	Mel Gibson	Danny Glover
7	Michael	John Travolta	Andie MacDowell
8	National Lampoon's Vacation	Chevy Chase	Beverly D'Angelo
9	Rocky	Sylvester Stallone	Talia Shire
10	Silence of the Lambs	Anthony Hopkins	Jodie Foster
11	The Hunt for Red October	Sean Connery	Alec Baldwin
12	The Terminator	Arnold Schwarzenegge	Michael Biehn
13	Titanic	Leonardo DiCaprio	Kate Winslet

The Process of Match-Joining

A traditional join is the process of combining rows from two or more tables (maximum of 256 tables) into a single row in a newly created table or query. The specific type of join that we will examine in this paper is known as a match join. A match-join combines rows from two or more tables into a single row in a new table, or query, according to the values found in a common column in each table. Its purpose is to bring data together to explore exciting insights into data relationships. The process consists of a matching process between a table's rows bringing together some or all of two or more tables contents, illustrated in Figure 1.

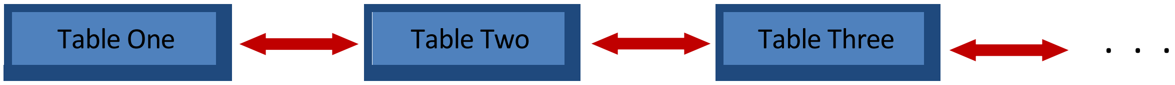


Figure 1. The Process of Joining Tables

The ability to define relationships between multiple tables and retrieve information based on these relationships is a powerful feature of the relational model. Joins are a data manipulation technique on a minimum of two tables, where a common column from each table is used for the purpose of combining the rows of data. The connecting column(s) should have the same column attributes and ***"like"*** values since the success of the process depends on these values. Unlike in a DATA step merge, the common column names do not have to be the same in a PROC SQL join.

Match-Join Features and Requirements

1. Portable to other vendor relational data base management systems.
2. Requires common column attributes in all tables.
3. Tables do not need to be sorted on common column.
4. Duplicate matching column is not automatically overlaid.
5. Results are automatically printed unless the NOPRINT option is specified.

Symmetrical Match-Joining

A traditional match-join process consists of combining rows in a symmetrical fashion from two or more tables. The result set from this type of matching process automatically eliminates unmatched rows and is referred to as the intersect (Movies_Actors) between the Movies and Actors tables, as shown in the Venn diagram in Figure 2.

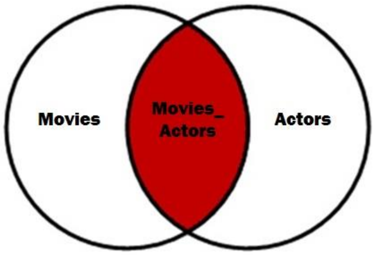


Figure 2. Venn Diagram – Match-Join

To illustrate the match-join process, the MOVIES and ACTORS tables are combined together using one or more common columns. In our example, TITLE is the common column found in both tables and is used to combine rows along with all selected columns (highlighted), as shown in Figure 3.

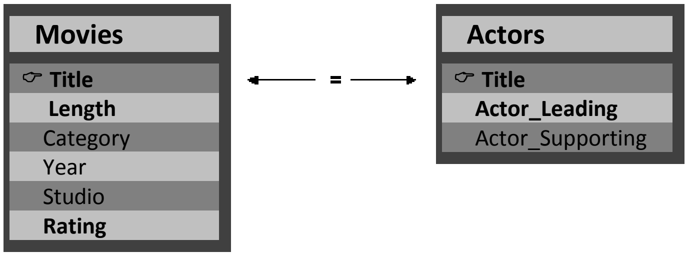


Figure 3. Match-Join using the MOVIES and ACTORS Tables

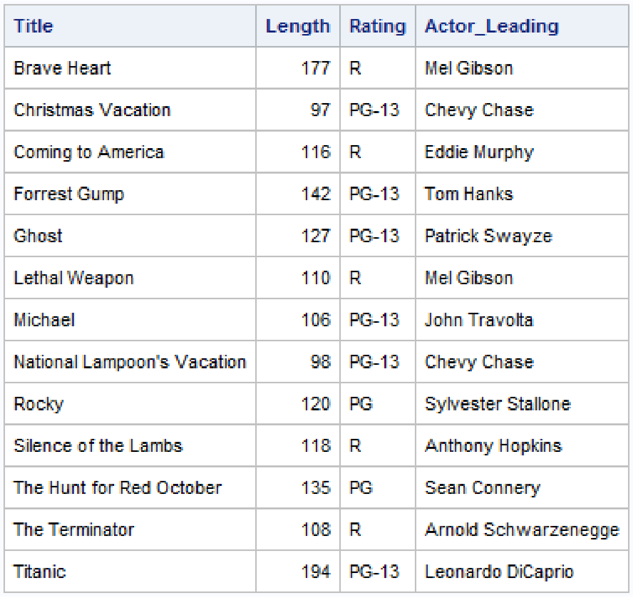
SAS SQL is invoked using the SQL procedure (aka, PROC SQL). Implemented as an interactive procedure, PROC SQL supports comprehensive data access, data manipulation, data management and reporting features; and remains active until turned off with a QUIT statement. Data access and manipulation is handled using with queries and the SELECT statement. SELECT informs SAS about which columns to capture in the result set, a required FROM-clause that identifies the SAS tables to read as input, and an optional WHERE-clause which tells SAS how to construct the subsetted results.

In the following read-only SELECT query an inner join query, or equijoin, result set is constructed in a WHERE-clause using the common column, TITLE, from the MOVIES and ACTORS tables. **Note:** The equal sign "=" comparison operator is specified to make the desired connection between the "key" column, TITLE, in the MOVIES and ACTORS tables. The result set of "matched" rows contains the columns, TITLE, LENGTH, RATING, and ACTOR_LEADING.

```
PROC SQL ;
  SELECT  MOVIES.TITLE,  LENGTH,  RATING,  ACTOR_LEADING
    FROM  MOVIES,  ACTORS
      WHERE MOVIES.TITLE  =  ACTORS.TITLE ;
QUIT ;
```

The result set from the equijoin SELECT query is shown below.

Results



Title	Length	Rating	Actor_Leading
Brave Heart	177	R	Mel Gibson
Christmas Vacation	97	PG-13	Chevy Chase
Coming to America	116	R	Eddie Murphy
Forrest Gump	142	PG-13	Tom Hanks
Ghost	127	PG-13	Patrick Swayze
Lethal Weapon	110	R	Mel Gibson
Michael	106	PG-13	John Travolta
National Lampoon's Vacation	98	PG-13	Chevy Chase
Rocky	120	PG	Sylvester Stallone
Silence of the Lambs	118	R	Anthony Hopkins
The Hunt for Red October	135	PG	Sean Connery
The Terminator	108	R	Arnold Schwarzenegge
Titanic	194	PG-13	Leonardo DiCaprio

Asymmetrical Match-Joining

A conventional join represents the combined rows from one table with rows in another symmetrically. But, occasionally rows need to be captured differently than in a conventional join. One approach, referred to as an asymmetrical type of join, is designed to preserve unmatched rows from one or both tables along with the matching rows.

Left Outer Join

A Left Outer join produces matched rows from two or more tables while preserving all unmatched rows from the first specified (left) table. A ***Left Outer join*** is illustrated by the shaded areas (Movies and Movies_Actors) in the Venn diagram, illustrated in Figure 4.

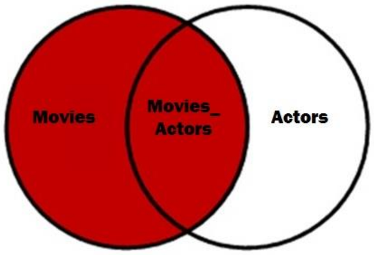


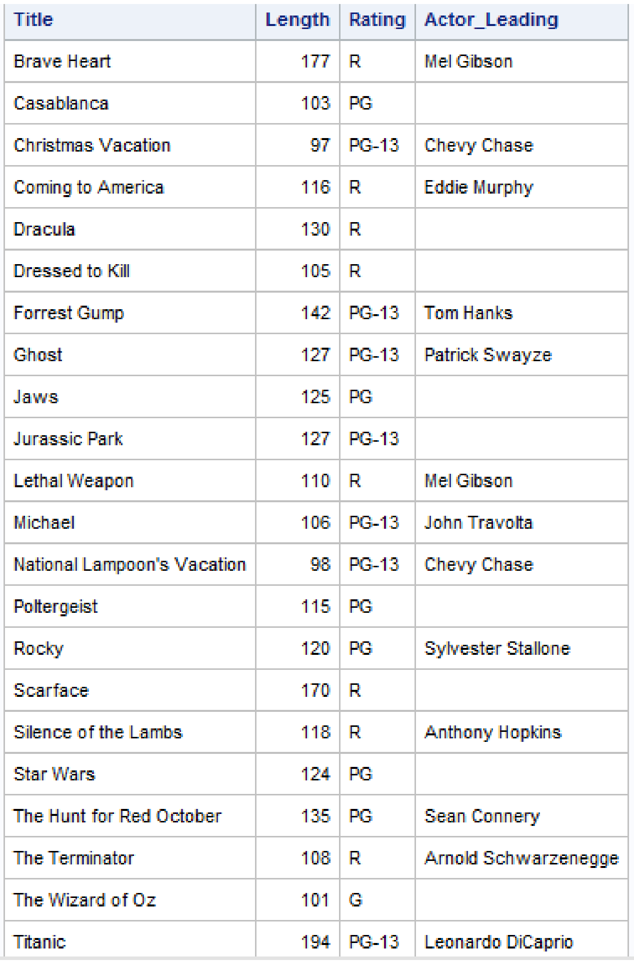
Figure 4. Venn Diagram – Left Outer Join

The join code, illustrated below, illustrates a left outer join construct. This read-only SELECT query specifies an ON-clause using the common column, TITLE, from the MOVIES and ACTORS tables to produce a result set of "matched" movies plus the preservation of all "unmatched" movies. **Note:** The equal sign "=" comparison operator is specified to make the desired connection between the "key" column, TITLE, in the MOVIES and ACTORS tables. The result set of "matched" rows contains the columns, TITLE, LENGTH, RATING, and ACTOR_LEADING.

```
PROC SQL ;
  SELECT  MOVIES.TITLE,  LENGTH,  RATING,  ACTOR_LEADING
   FROM  MOVIES
        LEFT JOIN
         ACTORS
     ON MOVIES.TITLE  =  ACTORS.TITLE ;
QUIT ;
```

The result set from the left outer join SELECT query is shown below.

Results



Title	Length	Rating	Actor_Leading
Brave Heart	177	R	Mel Gibson
Casablanca	103	PG	
Christmas Vacation	97	PG-13	Chevy Chase
Coming to America	116	R	Eddie Murphy
Dracula	130	R	
Dressed to Kill	105	R	
Forrest Gump	142	PG-13	Tom Hanks
Ghost	127	PG-13	Patrick Swayze
Jaws	125	PG	
Jurassic Park	127	PG-13	
Lethal Weapon	110	R	Mel Gibson
Michael	106	PG-13	John Travolta
National Lampoon's Vacation	98	PG-13	Chevy Chase
Poltergeist	115	PG	
Rocky	120	PG	Sylvester Stallone
Scarface	170	R	
Silence of the Lambs	118	R	Anthony Hopkins
Star Wars	124	PG	
The Hunt for Red October	135	PG	Sean Connery
The Terminator	108	R	Arnold Schwarzenegge
The Wizard of Oz	101	G	
Titanic	194	PG-13	Leonardo DiCaprio

Right Outer Join

A Right Outer join produces matched rows from two or more tables while preserving all unmatched rows from the second specified (right) table. A ***Right Outer join*** is illustrated by the shaded areas (Movies and Movies_Actors) shown by the Venn diagram, illustrated in Figure 5.

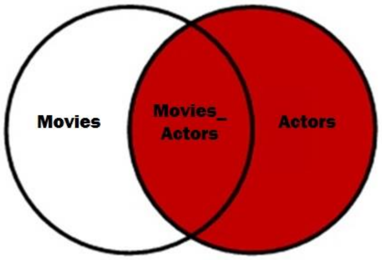


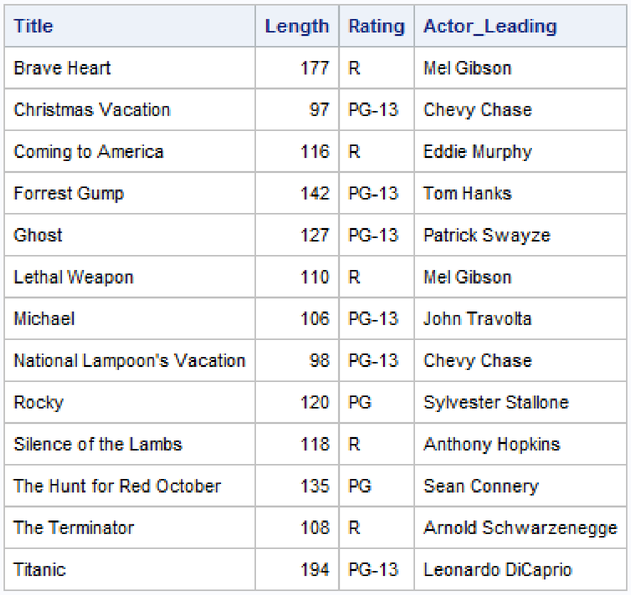
Figure 5. Venn Diagram – Right Outer Join

The join code, illustrated below, illustrates a right outer join construct. This read-only SELECT query specifies an ON-clause using the common column, TITLE, from the MOVIES and ACTORS tables to produce a result set of “matched” movies and actors, plus the preservation of all “unmatched” actors. **Note:** The equal sign “=” comparison operator is specified to make the desired connection between the “key” column, TITLE, in the MOVIES and ACTORS tables. The result set of “matched” rows contains the columns, TITLE, LENGTH, RATING, and ACTOR_LEADING.

```
PROC SQL ;
  SELECT  MOVIES.TITLE,  LENGTH,  RATING,  ACTOR_LEADING
    FROM  MOVIES
          RIGHT JOIN
           ACTORS
     ON MOVIES.TITLE  =  ACTORS.TITLE ;
QUIT ;
```

The result set from the right outer join SELECT query is shown below.

Results



Title	Length	Rating	Actor_Leading
Brave Heart	177	R	Mel Gibson
Christmas Vacation	97	PG-13	Chevy Chase
Coming to America	116	R	Eddie Murphy
Forrest Gump	142	PG-13	Tom Hanks
Ghost	127	PG-13	Patrick Swayze
Lethal Weapon	110	R	Mel Gibson
Michael	106	PG-13	John Travolta
National Lampoon's Vacation	98	PG-13	Chevy Chase
Rocky	120	PG	Sylvester Stallone
Silence of the Lambs	118	R	Anthony Hopkins
The Hunt for Red October	135	PG	Sean Connery
The Terminator	108	R	Arnold Schwarzenegge
Titanic	194	PG-13	Leonardo DiCaprio

Conclusion

The SAS SQL procedure, (aka, PROC SQL), is a powerful relational data base management system (RDBMS) language for SAS users to explore and use in a variety of application situations. This paper presented a brief introduction to the exciting world of PROC SQL joins, what a join is, illustrations of the various join techniques, and how PROC SQL can be used to join two or more tables. You are encouraged to explore these and other techniques to make your SAS experience a more rewarding and exciting one.

References

Lafler, Kirk Paul (2013). *PROC SQL: Beyond the Basics Using SAS, Second Edition*, SAS Institute Inc., Cary, NC, USA.

Lafler, Kirk Paul (2012), *"Exploring DATA Step Merges and PROC SQL Joins,"* Proceedings of the 2012 SAS Global Forum (SGF) Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2011), *"Exploring DATA Step Merges and PROC SQL Joins,"* Proceedings of the 2011 PharmaSUG Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2010), *"DATA Step and PROC SQL Programming Techniques,"* Ohio SAS Users Group (OSUG) 2010 One-Day Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2009), *"DATA Step and PROC SQL Programming Techniques,"* South Central SAS Users Group (SCSUG) 2009 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2009), *"DATA Step versus PROC SQL Programming Techniques,"* Sacramento Valley SAS Users Group 2009 Meeting, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul, Advanced SAS® Programming Tips and Techniques; Software Intelligence Corporation, Spring Valley, CA, USA; 1987-2007.

Lafler, Kirk Paul (2007), *"Undocumented and Hard-to-find PROC SQL Features,"* Proceedings of the PharmaSUG 2007 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul and Ben Cochran (2007), *"A Hands-on Tour Inside the World of PROC SQL Features,"* Proceedings of the SAS Global Forum (SGF) 2007 Conference, Software Intelligence Corporation, Spring Valley, CA, and The Bedford Group, USA.

Lafler, Kirk Paul (2006), *"A Hands-on Tour Inside the World of PROC SQL,"* Proceedings of the 31st Annual SAS Users Group International Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

Lafler, Kirk Paul (2005), *"Manipulating Data with PROC SQL,"* Proceedings of the 30th Annual SAS Users Group International Conference, Software Intelligence Corporation, Spring Valley, CA, USA.

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

Acknowledgments

The author thanks Dave Foster, Rapid Fire Section Chair, for accepting my abstract and paper; Michael G. Wilson, MWSUG 2015 Academic Chair, David Bruckner, MWSUG 2015 Operations Chair, the MidWest SAS Users Group (MWSUG) Executive Board, and SAS Institute for organizing and supporting a great conference!

Trademark Citations

About the Author

Kirk Paul Lafler has been using SAS since 1979 and is consultant and founder of Software Intelligence Corporation. He is a SAS Certified Professional, provider of IT consulting services, trainer to SAS users around the world, mentor, and sasCommunity.org emeritus Advisory Board member. As the author of six books including Google® Search Complete! (Odyssey Press. 2014) and PROC SQL: Beyond the Basics Using SAS, Second Edition (SAS Press. 2013); Kirk has written more than five hundred papers and articles; been an Invited speaker and trainer at five hundred-plus SAS International, regional, special-interest, local, and in-house user group conferences and meetings; and is the recipient of 23 "Best" contributed paper, hands-on workshop (HOW), and poster awards.

Comments and suggestions can be sent to:

Kirk Paul Lafler
Senior SAS® Consultant, Application Developer, Data Scientist, Trainer and Author
Software Intelligence Corporation
E-mail: KirkLafler@cs.com
LinkedIn: http://www.linkedin.com/in/KirkPaulLafler
Twitter: @sasNerd