

Kirk's Korner



of Quick & Simple Tips

@sasNerd

Identifying FIRST. and LAST. Rows in By-Groups with PROC SQL

A popular DATA step technique that millions of SAS users need and find useful is the ability to identify the first and last observation in by-groups. SAS performs by-group processing by creating two temporary variables for each BY variable: a FIRST.variable and a LAST.variable. These temporary variables are populated with a value of 0 or 1 depending on the placement of each observation in the by-group. For example, when an observation is first in a by-group, SAS automatically assigns a value of 1 to the FIRST.variable. For all other observations in the by-group, SAS assigns a value of 0 to FIRST.variable. Similarly, if an observation is last in a by-group, SAS assigns a value of 1 to LAST.variable. For all other observations in the by-group a value of 0 is assigned to LAST.variable. It should be noted that these two temporary variables are available for conditional processing in the DATA step, and are not automatically added to the output data set.

It should be noted that although the ability to identify FIRST. and LAST. observations in by-groups has been available to DATA step users for a long time, the same is not true for traditional SQL users. Essentially this meant that whenever an SQL user needed the capability to identify FIRST. and LAST. rows in by-groups, they had little choice but to use the DATA step programming technique. This typically was not an acceptable approach for the SQL purist, so I knew I had to do something.

Over the next three years I spent much of my free time researching and experimenting with every conceivable coding approach I could muster. Finally, I came up with an acceptable SQL solution that could emulate the features found in the stalwart DATA step approach. The result is pure SQL code that uses a subquery construct. Now, SQL users everywhere, and even users working with many of the leading RDBMS', have a way to identify the FIRST. and LAST. rows in by-groups.

This tip illustrates the FIRST. and LAST. coding technique using a single table MOVIES, as shown below. The Movies table consists of 22 rows and six columns: title, category, studio, and rating are defined as character, and length and year are defined as numeric. After the FIRST. and LAST. rows are identified, the rows are then concatenated.

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	Pollergeist	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

Identifying FIRST. Row in By-groups

SQL Code

```
/*  
** ROUTINE: FIRST-BY-GROUP-ROWS **  
** PURPOSE: Derive the first (min) row within each by-group using a subquery. **  
*/  
proc sql ;  
  create table first_bygroup_rows as  
  select rating,  
         title,  
         'FirstRow' as ByGroup  
  from movies M1  
  where title =  
         (select min(title)  
          from movies M2  
          where M1.rating = M2.rating)  
  order by rating, title ;
```

FIRST. Results

Rating	Title	ByGroup
G	The Wizard of Oz	FirstRow
PG	Casablanca	FirstRow
PG-13	Christmas Vacation	FirstRow
R	Brave Heart	FirstRow

Identifying LAST. Row in By-groups

SQL Code

```
/*  
** ROUTINE: LAST-BY-GROUP-ROWS **  
** PURPOSE: Derive the last (max) row within each by-group using a subquery. **  
*/  
create table last_bygroup_rows as  
  select rating,  
         title,  
         'LastRow' as ByGroup  
  from movies M1  
  where title =  
         (select max(title)  
          from movies M2  
          where M1.rating = M2.rating)  
  order by rating, title ;
```

LAST. Results

Rating	Title	ByGroup
G	The Wizard of Oz	LastRow
PG	The Hunt for Red October	LastRow
PG-13	Titanic	LastRow
R	The Terminator	LastRow

Concatenating FIRST. and LAST. Rows

SQL Code

```
/* *****  
/** ROUTINE: CONCATENATE-FIRST-LAST-BY-GROUP-ROWS **/  
/** PURPOSE: Concatenate the results from the first (min) row and last (max) **/  
/**           row within each by-group, and print. **/  
/* *****  
create table first_last_rows as  
  select rating, title, bygroup  
         from first_bygroup_rows  
  
  UNION ALL  
  
  select rating, title, bygroup  
         from last_bygroup_rows ;  
  
select * from first_last_rows ;  
quit ;
```

FIRST. and LAST. Results Combined

Rating	Title	ByGroup
G	The Wizard of Oz	FirstRow
PG	Casablanca	FirstRow
PG-13	Christmas Vacation	FirstRow
R	Brave Heart	FirstRow
G	The Wizard of Oz	LastRow
PG	The Hunt for Red October	LastRow
PG-13	Titanic	LastRow
R	The Terminator	LastRow

For More Information:

If readers are interested in finding out how to identify the BETWEEN. rows, please send an email to KirkLafler@cs.com and I'll be happy to send the code. Readers can also find many more application programming techniques in my SAS Press book, **PROC SQL: Beyond the Basics Using SAS, Second Edition**.

About The Author



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

Kirk Paul Lafler

Senior SAS Consultant, Application Developer, Data Scientist, Educator and Author
Software Intelligence Corporation

E-mail: KirkLafler@cs.com

LinkedIn: <http://www.linkedin.com/in/KirkPaulLafler>

Twitter: @sasNerd

[Google Search Complete!](#)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Google, Google Search, and other Google products and services are the registered trademarks of Google Inc., Mountain View, California, USA. Other brand and product names are trademarks of their respective companies.