

Presenter Bio



Kirk has been a SAS software user since 1979, and has worked as a consultant, application developer, programmer, data analyst, educator and author. Currently, Kirk is a lecturer and adjunct professor at San Diego State University; an advisor and adjunct professor at the University of California San Diego Extension; and teaches dozens of SAS, SQL, R and Python courses, seminars, workshops, and webinars. As the author of several books including PROC SQL: Beyond the Basics Using SAS, Third Edition (SAS Press. 2019) along with hundreds of papers and articles on a variety of SAS topics; Kirk has served as an Invited speaker, educator, keynote and section leader at SAS conferences and e-conferences worldwide; and is the recipient of 25 “Best” contributed paper, hands-on workshop (HOW), and poster awards.

Presentation Topics

MSGLEVEL=I SAS System Option

PROC SQL Join Algorithms

_METHOD PROC SQL Option

SQL Query Optimization Techniques

Conclusion



Under the Hood: The Mechanics of Various SQL Query Optimization Techniques

a presentation by

Kirk Paul Lafler

<https://www.linkedin.com/in/KirkPaulLafler/>

KirkLafler@cs.com

@sasNerd



Copyright © 2020

Kirk Paul Lafler and @sasNerd

All rights reserved.

SAS is the registered trademark of SAS Institute Inc., Cary, NC, USA.

All other company and product names mentioned are used for identification purposes only and may be trademarks of their respective owners.

Presentation Topics

MSGLEVEL=I SAS System Option

PROC SQL Join Algorithms

_METHOD PROC SQL Option

SQL Query Optimization Techniques

Conclusion

Presentation Topics



MSGLEVEL=I
SAS
System
Option

Presentation Topics



MSGLEVEL=I
SAS
System
Option



PROC SQL
Join
Algorithms

Presentation Topics



MSGLEVEL=I
SAS
System
Option



PROC SQL
Join
Algorithms



_METHOD
PROC SQL
Option

Presentation Topics



MSGLEVEL=I
SAS
System
Option



PROC SQL
Join
Algorithms



_METHOD
PROC SQL
Option



SQL
Query
Optimization
Techniques

Example Table #1

Movies

	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

Example Table #2

Actors

	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 Schwarzenegger	Michael Biehn
13	Titanic	Leonardo DiCaprio	Kate Winslet

MSGLEVEL=I

SAS System

Option

MSGLEVEL=I SAS System Option

By specifying the **OPTIONS MSGLEVEL=I;** SAS System option, helpful informational notes describing index usage, sort utilities, and merge processing are displayed on the SAS Log.

```
OPTIONS MSGLEVEL=I ;
```

```
PROC SQL ;
```

```
    SELECT M.TITLE, RATING, LENGTH, ACTOR_LEADING
```

```
    FROM MOVIES M,
```

```
         ACTORS A
```

```
    WHERE M.TITLE = A.TITLE
```

```
    AND RATING = 'PG' ;
```

```
QUIT ;
```

MSGLEVEL=I SAS System Option

SAS Log Results:

```
OPTIONS MSGLEVEL=I ;
```

```
PROC SQL ;  
    SELECT M.TITLE, RATING, LENGTH, ACTOR_LEADING  
    FROM MOVIES M,  
    ACTORS A  
    WHERE M.TITLE = A.TITLE  
    AND RATING = 'PG' ;
```

INFO: Index Rating selected for WHERE clause optimization.

```
QUIT ;
```

Presentation Topics

MSGLEVEL=I SAS System Option

PROC SQL Join Algorithms

_METHOD PROC SQL Option

SQL Query Optimization Techniques

Conclusion

PROC SQL Join Algorithms

PROC SQL

Join Algorithms

4 Join Algorithms are available to the SQL Optimizer:

- **Nested (or Step) Loop**
- **Sort-Merge**
- **Index**
- **Hash**

PROC SQL Join Algorithms

A nested (or step) loop join algorithm may be selected by the SQL optimizer when:

- **Processing small tables of data;**
- **One table is considerably smaller than the other table;**
- **Join condition does not contain an equality condition;**
- **Sort-merge, index, and hash join algorithms have been eliminated from consideration.**

PROC SQL

Join Algorithms

A sort-merge join algorithm is selected most often by the SQL optimizer when:

- **Processing small to medium size tables of data;**
- **Nested (or Step) loop, index, and hash join algorithms have been eliminated from consideration.**

PROC SQL

Join Algorithms

An index join algorithm may be selected by the SQL optimizer when:

- **It is determined that using an existing simple or composite index will improve the performance of the join relation;**
- **Nested (or Step) loop, sort-merge, and hash join algorithms have been eliminated from consideration.**

PROC SQL

Join Algorithms

A hash join algorithm may be selected by the SQL optimizer when:

- **Sufficient memory is available to process at least the smaller table into memory;**
- **The MEMSIZE= and BUFFERSIZE= options are set optimally.**

Presentation Topics

MSGLEVEL=I SAS System Option

PROC SQL Join Algorithms

_METHOD PROC SQL Option

SQL Query Optimization Techniques

Conclusion

_METHOD PROC SQL Option

`_METHOD` PROC SQL Option

A **`_METHOD`** option can be specified on the PROC SQL statement to display the processing hierarchy (or execution plan) of a query. The results of the **`_METHOD`** option are displayed with codes on the SAS Log.

_METHOD PROC SQL Option

The list of codes and descriptions for the _METHOD option are:

Code	Description
sqxcrt	Create table statement
sqxslct	Select statement
sqxjsl	Step loop join (Cartesian Product) join
sqxjm	Merge join
sqxjndx	Index join
sqxjhsh	Hash join
sqxsort	Sort
sqxsrc	Table source
sqxfil	Filter rows
sqxsumg	Summary statistics with GROUP BY
sqxsumn	Summary statistics with no GROUP BY

_METHOD PROC SQL Option

Program Example using _METHOD:

```
PROC SQL _METHOD ;  
    SELECT M.TITLE, RATING, ACTOR_LEADING  
    FROM MOVIES M,  
         ACTORS A  
    WHERE M.TITLE = A.TITLE  
         AND RATING = 'PG' ;  
QUIT ;
```

_METHOD PROC SQL Option

SAS Log Results:

NOTE: SQL execution methods chosen are:

sqxslct

sqxjhsh

sqxsrc(MOVIES)

sqxsrc(ACTORS)

Using MSGLEVEL= and _METHOD

Program Example using MSGLEVEL= and _METHOD:

```
OPTIONS MSGLEVEL=I ;  
PROC SQL _METHOD ;  
    SELECT M.TITLE, RATING, ACTOR_LEADING  
    FROM MOVIES M,  
    ACTORS A  
    WHERE M.TITLE = A.TITLE  
    AND RATING = 'PG' ;  
QUIT ;
```


Using MSGLEVEL= and _METHOD

SAS Log Results:

```
OPTIONS MSGLEVEL=I ;  
PROC SQL _METHOD ;  
  SELECT M.TITLE, RATING, LENGTH, ACTOR_LEADING  
  FROM MOVIES M,  
       ACTORS A  
  WHERE M.TITLE = A.TITLE  
        AND RATING = 'PG' ;
```

NOTE: SQL execution methods chosen are:

sqxslct

sqxjhsh

sqxsrc(MOVIES)

sqxsrc(ACTORS)

INFO: Index Rating selected for WHERE clause optimization.

```
QUIT ;
```

Presentation Topics

MSGLEVEL=I SAS System Option

PROC SQL Join Algorithms

_METHOD PROC SQL Option

SQL Query Optimization Techniques

Conclusion

SQL Query Optimization Techniques

~ ~ ~ ~ ~

System Options

STIMER / FULLSTIMER

~ ~ ~ ~ ~

System Options **STIMER / FULLSTIMER**

- Turn on the SAS system option, STIMER and/or FULLSTIMER;
- **OPTIONS STIMER < or > FULLSTIMER ;**
- Run and Measure performance results;
- Evaluate performance results;
- Modify a query's execution to achieve an optimal balance between competing computer resources.

Resource Definitions for STIMER / FULLSTIMER

- **CPU** – time used to decode and execute programs;
- **I/O** – time used to perform read / write operations;
- **Memory** – location where data is stored while being processed;
- **Data storage** – secondary storage where data is kept;
- **Programming** – human element, techniques, and implementation of coding constructs.

Resource Definitions for STIMER / FULLSTIMER

■ Options STIMER;

NOTE: DATA statement used (Total process time):

real time 0.04 seconds

cpu time 0.03 seconds

■ Options FULLSTIMER;

NOTE: DATA statement used (Total process time):

real time 0.05 seconds

user cpu time 0.03 seconds

system cpu time 0.04 seconds

memory 661.23k

OS Memory 6768.00k

Timestamp 09/16/2012 11:17:19 PM

~ ~ ~ ~ ~

SELECT Clause

Syntax / Execution Order

~ ~ ~ ~ ~

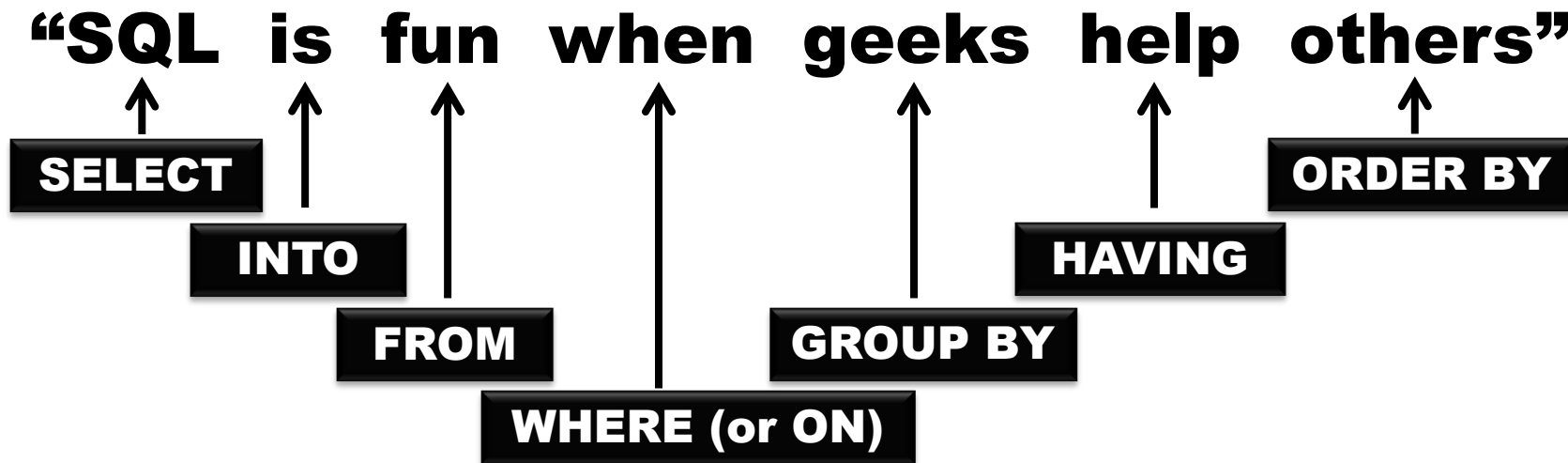
SELECT Clause

Syntax / Execution Order

The SELECT statement's purpose is to retrieve (or read) data from one, or more, underlying tables (or views). Although the SELECT statement supports multiple clauses, only one clause is required – the FROM clause. All remaining clauses are optional and only specified as needed.

SELECT Clause Syntax Order

To help remember the specific order of the SELECT statement's clauses, recite this phrase:



SELECT Clause Syntax Order

The syntax order of the SELECT clauses are:

```

PROC SQL ;
  SELECT . . .
    INTO . . .
      FROM . . .
        WHERE < OR > ON . . .
          GROUP BY . . .
            HAVING . . .
              ORDER BY . . . ;
QUIT ;

```

SELECT Clause Execution Order

Execution Order	
1. FROM	The first clause executed in a query is the FROM clause. It's a required clause with the purpose of determining the working set of data that is being queried (i.e., variable names, variable type, number of rows, and other important information).
2. INTO	The INTO clause is used to create one or more macro variables where the values can be used to manipulate data.
3. ON	The ON clause is used to subset rows of data based on the condition(s) specified, and rows that aren't satisfied by the condition(s) are discarded.
4. WHERE	The WHERE clause is used to subset rows of data based on the condition(s) specified, and rows that aren't satisfied by the condition(s) are discarded.
5. GROUP BY	The GROUP BY clause takes the rows that were subset with the WHERE clause and grouped based on common values in the column specified in the GROUP BY clause.
6. HAVING	The HAVING clause applies the condition(s) to the grouped rows specified in the GROUP BY clause, and any grouped rows that aren't satisfied by the condition(s) are discarded.
7. SELECT	Expressions specified in the SELECT statement are processed.
8. ORDER BY	The ORDER BY clause sorts the rows of data in either ascending (default) or descending order.

~ ~ ~ ~ ~

Logic Construction and Optimization

~ ~ ~ ~ ~

Constructing Efficient Logic

- Logic conditions affect processing costs;
- The SQL optimizer evaluates a series of “ANDed” expressions in a WHERE (or ON) from left to right;
- A chain of “ANDed” conditions should be specified with the most restrictive expression first;
- As a result, fewer resources are spent by bypassing rows that do not satisfy the first conditional value in the WHERE (or ON) clause.

~ ~ ~ ~ ~

Indexes and Optimization

~ ~ ~ ~ ~

Default SAS and SQL Processing

A database table:

- By default, is traversed in sequential order (top to bottom);
- When a table is small to medium size (based on the number of observations or rows) the costs of processing rows of data sequentially may not present a problem;
- But, as a table grows in size, the costs of processing a table sequentially may become prohibitively expensive.

Sequential versus Index Processing

Sequential Processing

Obs	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

Index Processing

Index
 Category =
 "Comedy"
 Rows =
 3,4,13

Obs	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

What is an Index

A database index:

- Is a data structure that is designed to help improve the speed of data retrieval operations on a database table;
- Can introduce additional CPU, I/O and storage costs.

Benefits of an Index

An index may be able to improve performance:

- **By providing faster and more efficient access to data subsets;**
- **By returning rows in ascending value order for BY-group processing without using the SORT procedure.**

Types of Indexes

Types of Indexes:

- **Simple Index – single column**
 - ✓ index is assigned the same name as the column
- **Composite Index – two or more columns**
 - ✓ index is assigned a unique user-defined name

Anatomy of an Index

- Possesses the same name as the SAS dataset;
- Stored in the same SAS library as its dataset;
- Represents a separate entity and has a member type of INDEX;
- A binary search is performed on the index file;
- Positions the index to the first qualified value that matches the WHERE clause expression using pointers.

Index Entries and Pointers

- An index file contains entries organized hierarchically;
- Entries are connected by pointers;
- Entries are organized in ascending value order;
- Entries contain the following information:
 - ✓ A unique value;
 - ✓ One or more unique observation(s), referred to as the record identifier (RID).

Index Entries and RID Pointers

- Entries representing the index file for RATING are:

Value	RID
G	21
PG	2, 9, 14, 15, 18, 19
PG-13	3, 7, 8, 10, 12, 13, 22
R	1, 4, 5, 6, 11, 16, 17, 20

Indexes and Costs

- An index must be maintained by the SAS System;
- An index is updated each time a table is modified (e.g., new row, deleted row, updated row);
- The SQL optimizer determines whether an index will be used.

Index Usage Rules

- Triggered with WHERE (or ON) clause processing;
- Avoid defining more indexes than are needed;
- If subset is large, then sequential access may be just as, or even more efficient;
- Avoid constructing indexes when the Data File Page Count from the CONTENTS procedure is less than **3**;
- If subset is **15%** or less then optimizer may use index.

Page Count PROC CONTENTS

```
proc contents data=movies  
details;  
run;
```

Engine/Host Dependent Information	
Data Set Page Size	8192
Number of Data Set Pages	2
First Data Page	1
Max Obs per Page	92
Obs in First Data Page	22
Index File Page Size	4096
Number of Index File Pages	4
Number of Data Set Repairs	0
File Name	C:\Workshops\Sample Data\movies.sas7bdat
Release Created	9.0101M0
Host Created	XP_HOME

Selecting Columns

- **Select column(s) that is/are used in queries containing WHERE (or ON) clause processing;**
- **The column(s) used in an index should select the fewest observations;**
- **When creating a composite index, the first column should be the most discriminating, the second column should be the next most discriminating, and so on.**

One-way Frequency Analysis with PROC FREQ

```
proc freq data=libref.movies ;  
  tables category ;  
run ;
```

The FREQ Procedure

Category	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Action	1	4.55	1	4.55
Action Adventure	4	18.18	5	22.73
Action Cops & Robber	2	9.09	7	31.82
Action Sci-Fi	2	9.09	9	40.91
Adventure	1	4.55	10	45.45
Comedy	3	13.64	13	59.09
Drama	3	13.64	16	72.73
Drama Mysteries	1	4.55	17	77.27
Drama Romance	2	9.09	19	86.36
Drama Suspense	1	4.55	20	90.91
Horror	2	9.09	22	100.00

One-way Frequency Analysis with PROC FREQ

```
proc freq data=libref.movies ;  
  tables rating ;  
run ;
```

The FREQ Procedure

Rating	Frequency	Percent	Cumulative Frequency	Cumulative Percent
G	1	4.55	1	4.55
PG	6	27.27	7	31.82
PG-13	7	31.82	14	63.64
R	8	36.36	22	100.00

Two-way Frequency Analysis with PROC FREQ

```
proc freq data=libref.movies ;
  tables category * rating ;
run ;
```

The FREQ Procedure

Frequency
Percent

Table of Category by Rating					
Category	Rating				Total
	G	PG	PG-13	R	
Action	0 0.00	0 0.00	1 4.55	0 0.00	1 4.55
Action Adventure	0 0.00	3 13.64	0 0.00	1 4.55	4 18.18
Action Cops & Robber	0 0.00	0 0.00	0 0.00	2 9.09	2 9.09
Action Sci-Fi	0 0.00	1 4.55	0 0.00	1 4.55	2 9.09
Adventure	1 4.55	0 0.00	0 0.00	0 0.00	1 4.55
Comedy	0 0.00	0 0.00	2 9.09	1 4.55	3 13.64
Drama	0 0.00	1 4.55	2 9.09	0 0.00	3 13.64
Drama Mysteries	0 0.00	0 0.00	0 0.00	1 4.55	1 4.55
Drama Romance	0 0.00	0 0.00	2 9.09	0 0.00	2 9.09
Drama Suspense	0 0.00	0 0.00	0 0.00	1 4.55	1 4.55
Horror	0 0.00	1 4.55	0 0.00	1 4.55	2 9.09
Total	1 4.55	6 27.27	7 31.82	8 36.36	22 100.00

Evaluating Indexes with **IDXWHERE=**

By specifying the **IDXWHERE=** data set option, SAS will ignore sequential processing and select the best available index to use.

```
OPTIONS MSGLEVEL=I ;
```

```
PROC SQL _METHOD ;
```

```
    SELECT TITLE, CATEGORY, RATING
```

```
    FROM libref.MOVIES (IDXWHERE=YES)
```

```
    WHERE RATING = 'G' ;
```

```
QUIT ;
```

Selecting an Index with **IDXNAME=**

By specifying the **IDXNAME=** data set option, SAS will ignore sequential processing and allow you to select the best available index to use.

```
OPTIONS MSGLEVEL=I ;  
PROC SQL _METHOD ;  
    SELECT TITLE, CATEGORY, RATING  
    FROM libref.MOVIES (IDXNAME=CATEGORY)  
    WHERE CATEGORY = 'Comedy' ;  
QUIT ;
```

Conclusion



MSGLEVEL=I
SAS
System
Option

Conclusion



MSGLEVEL=I
SAS
System
Option



PROC SQL
Join
Algorithms

Conclusion



MSGLEVEL=I
SAS
System
Option



PROC SQL
Join
Algorithms



_METHOD
PROC SQL
Option

Conclusion



MSGLEVEL=I
SAS
System
Option



PROC SQL
Join
Algorithms

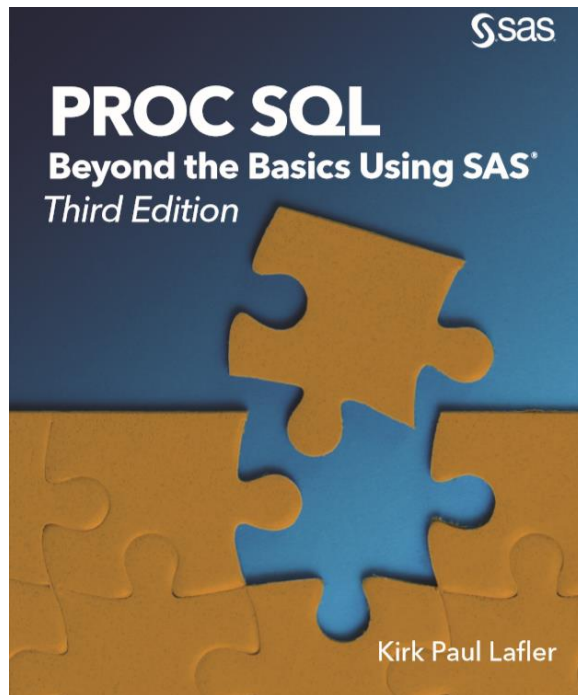


_METHOD
PROC SQL
Option



SQL
Query
Optimization
Techniques

PROC SQL: Beyond the Basics Using SAS, 3rd Ed.



Intermediate and Advanced PROC SQL content including fuzzy matching and data-driven programming techniques

Available on Amazon.com and Online Book Stores everywhere!

Cutting Edge Training from Kirk Paul Lafler

Elevate Your SAS® Skills with Cutting-Edge Training

Kirk Paul Lafler offers dozens of popular cutting-edge SAS® instructor-led, courses, e-courses, hands-on workshops and webinars around the world. Whether you're a beginner, intermediate or advanced user, you'll find quality training solutions to enhance and elevate your SAS skills and knowledge. Attendees learn practical and results-oriented techniques and solutions using real-world examples. Training courses are affordably priced, comprehensive and immediately available. For more information, please send an email to KirkLafler@cs.com.

Kirk Paul Lafler

<https://www.linkedin.com/in/KirkPaulLafler/>

KirkLafler@cs.com

Thank You for Attending!

Questions?

a presentation by

Kirk Paul Lafler

<https://www.linkedin.com/in/KirkPaulLafler/>

KirkLafler@cs.com

@sasNerd

