

Quicker Than Merge?
Kirby Cossey
State Auditor's Office, Austin, Texas

ABSTRACT

How many times do you need to extract a few records from a large dataset? This paper compares four techniques: traditional merge, using SAS to write a where statement for you, the key= option on a set statement, and SQL join.

INTRODUCTION

We have 11 monthly files ranging from 7,172,501 observations to 10,198,223 with 97 fields and a record length of 499. The size of the compressed SAS files were 2.3 gig to 3.2 gig. Several fields were indexed, including PCN. A small file containing 11182 unique PCN values exist that all corresponding records from the 11 large files. The small file also has the PCN variable indexed. The challenge is to retrieve this set of records in a reasonable time and develop a reasonable methodology for future such extractions for other fields. This was being done on a mainframe but was moved to a SAS Server connected to a SAN with 232 gig. 358,512 records were extracted from the 89,632,044 records in the 11 files.

TRADITIONAL MERGE ON MAINFRAME

Each month took from 2:25:22.48 to 3:48:41.54 in real time for a total of over 33 hours. The CPU time varied from 5:02.63 to 6:58.38 for a total of over 64 minutes of CPU time.

TRADITIONAL MERGE ON SAS SERVER

With nothing else running on the server, each month took from 52:16.60 to 1:56:32.04 in real time for a total of over 16 hours. The CPU time varied from 3:11.60 to 4:00.48 for a total of over 57 minutes of CPU time.

```

1      %MACRO MONTH(MONTH1,MONTH2);
2      data &Month2; merge E2.All&Month2(In=A) E.UniquePCNs(In=B);
3          By PCN;
4          If A & B;
5      Run;
6      %MEND;
7      %MONTH(Sep,Sep00); Run;
8      %MONTH(Oct,Oct00); Run;
9      %MONTH(Nov,Nov00); Run;
10     %MONTH(Dec,Dec00); Run;
11     %MONTH(Jan,Jan01); Run;
12     %MONTH(Feb,Feb01); Run;
13     %MONTH(Mar,Mar01); Run;
14     %MONTH(Apr,Apr01); Run;
15     %MONTH(May,May01); Run;
16     %MONTH(Jun,Jun01); Run;
17     %MONTH(Jul,Jul01); Run;
```

Lines 1 through 4 are the SAS code encapsulated in a SAS Macro named MONTH. The SAS Macro is run eleven times, one for each month of data.

SAS WRITES THE WHERE STATEMENT ON THE MAINFRAME

Each month took from 5:57.56 to 8:21.72 in real time for a total of over 74 minutes. The CPU time varied from 48.87 to 69.18 seconds for a total of over 10 minutes of CPU time.

SAS WRITES THE WHERE STATEMENT ON SAS SERVER

With nothing else running on the server, each month took from 42.17 to 5:49.98 in real time for a total of over 39 minutes. The CPU time varied from 0.86 to 69.87 seconds for a total of over 9 minutes of CPU time.

```
1      Filename List 'E:\PProc LT 9\List.txt'; run;

2      Data _Null_; File List; Set E.UniquePCNs END=EOF;
3          By PCN;
4          If _N_ = 1 Then Put "Where PCN IN (";
5          If EOF Then Do;
6              Put @1 "" @2 PCN @11 "" ;
7              Put ");";
8          End;
9          Else Put @1 "" @2 PCN @11 ", ";
10     Run;

%MACRO Month(Month1,Month2);
11     data &Month2;
12         Set Data.All&Month2;
13         %INCLUDE List;
14     Run;

%MEND;

%MONTH(Sep,Sep00); Run;
%MONTH(Oct,Oct00); Run;
%MONTH(Nov,Nov00); Run;
%MONTH(Dec,Dec00); Run;
%MONTH(Jan,Jan01); Run;
%MONTH(Feb,Feb01); Run;
%MONTH(Mar,Mar01); Run;
%MONTH(Apr,Apr01); Run;
%MONTH(May,May01); Run;
%MONTH(Jun,Jun01); Run;
%MONTH(Jul,Jul01); Run;
```

Lines 1 through 10 read a SAS dataset named E.UniquePCNs by a field named PCN and write a WHERE statement to a text file named List.txt in a folder named 'E:\PProc LT 9\.

Line 4 will execute the 1st time through the E.UniquePCNs dataset. Line 4 will write the start of the WHERE statement. Lines 5 through 8 will execute when the last record is read. Lines 7 through 8 write the end of the where statement. Line 9 will execute for all records except the last record. Lines 6 and 8 write the PCN value inside of quotes. Line 9 puts the comma after the PCN value for the WHERE list. Line 6 does not put a comma since it is the last value in the WHERE list. Lines 11 through 14 are the SAS code encapsulated in a SAS Macro named MONTH. The "%INCLUDE List" on line 13 brings in the text file named List.txt that contains the WHERE statement. The SAS Macro is run eleven times, one for each month of data.

KEY= SET OPTION ON THE MAINFRAME

Each month took from 1:28.41 to 2:17.51 in real time for a total of over 19 minutes. The CPU time varied from 1.30 to 1.85 seconds for a total of over 16 seconds of CPU time.

KEY= SET OPTION ON THE SERVER

With nothing else running on the server , each month took from 0:15.73 to 0:25.23 in real time for a total of over 3 minutes. The CPU time varied from 0:00.78 to 0:11.30 seconds for a total of over 11 seconds of CPU time.

This complicated use of the KEY= method will return multiple records from the master dataset.

```
%MACRO MONTH(MONTH1,MONTH2);
1      data &Month2; set E.UniquePCNs;
2          do until(_iorc_=%sysrc(_dsenom));
3              set E2.Data&Month2 key=PCN;
4              select(_iorc_);
5              when (%sysrc(_sok)) do;
6                  output;
7              end;
8              when (%sysrc(_dsenom)) do;
9                  _error_=0;
10                 DELETE;
11             end;
12             otherwise;
13             end;
14             end;
%MEND;
%MONTH(Sep,Sep00); Run;
%MONTH(Oct,Oct00); Run;
%MONTH(Nov,Nov00); Run;
%MONTH(Dec,Dec00); Run;
%MONTH(Jan,Jan01); Run;
%MONTH(Feb,Feb01); Run;
%MONTH(Mar,Mar01); Run;
%MONTH(Apr,Apr01); Run;
%MONTH(May,May01); Run;
%MONTH(Jun,Jun01); Run;
%MONTH(Jul,Jul01); Run;
```

Lines 1 through 14 use the KEY= dataset option. Line 1 sets dataset E.UniquePCNs. Line 3 sets the dataset E2.Data&Month2. Line 3 says "key= PCN". PCN is the indexed field for the lookup. Only the dataset names and index need to be changed.

Lines 2 and 4 - _iorc_ = 'input/output return code'. When it's equal to 0, SAS did not find a match. When ne 0, it found a matching observation.

Lines 2, 5, and 8 - %sysrc = returns a system error number.

Lines 2, and 8 - _dsenom = means no matching observation was found in the master dataset.

Line 5 - _sok = the i/o operation was successful.

SQL ON THE MAINFRAME

Each month took from 0:25.08 to 2:29.05 in real time for a total of over 19 minutes. The CPU time varied from 0:01.67 to 0:02.23 seconds for a total of over 20 seconds of CPU time.

SQL ON THE SERVER

With nothing else running on the server , each month took from 0:27.06 to 0:56.18 in real time for a total of over 8 minutes. The CPU time varied from 0:00.85 to 0:01.97 seconds for a total of over 17 seconds of CPU time.

Using files not sorted or indexed increased the time somewhat. With nothing else running on the server , each month took from 0:19.48 to 6:59.51 in real time for a total of over 43 minutes. The CPU time varied from 0:13.19 to 0:24.36 seconds for a total of over 3 minutes of CPU time.

```
%MACRO MONTH(MONTH1,MONTH2);
1      PROC SQL;
2      CREATE TABLE &Month2 AS
3      SELECT S.*
4      FROM DATA.UniquePCNs as BD,
5      DATA&MONTH1..dataedit&Month2 as S
6      WHERE BD.pcn = S.pcn;
7      QUIT;
%MEND;
%MONTH(Sep,Sep00); Run;
%MONTH(Oct,Oct00); Run;
%MONTH(Nov,Nov00); Run;
%MONTH(Dec,Dec00); Run;
%MONTH(Jan,Jan01); Run;
%MONTH(Feb,Feb01); Run;
%MONTH(Mar,Mar01); Run;
%MONTH(Apr,Apr01); Run;
%MONTH(May,May01); Run;
%MONTH(Jun,Jun01); Run;
%MONTH(Jul,Jul01); Run;
```

Lines 1 through 7 are the actual SQL code. Line 4 uses the DATA.UniquePCNs dataset. Line 5 uses the DATA&MONTH1..dataedit&Month2 dataset. Notice the .. used between DATA&MONTH1 and dataedit&Month2. This is required because DATA&MONTH1 ends with a macro variable. Line 6 performs a join based on the common indexed field PCN. The fields do not require the datasets to be sorted or indexed but will run quicker if they are. The common fields do not need to have the same name in SQL.

CONCLUSION

SQL gives the most flexibility when datasets are not sorted or indexed. The KEY= method works fastest when the fewest records are needed.

On the mainframe we went from
33 **hours** using the traditional merge to
74 **minutes** using SAS to write the Where statement to
19 **minutes** using the Key = option on the set statement to
19 **minutes** using SQL join.

On the SAS Server we went from
11 **hours** using the traditional merge to
39 **minutes** using SAS to write the Where statement to
9 **minutes** using the Key = option on the set statement to
8 **minutes** using SQL join.

	Mainframe Real Time	Mainframe CPU Time	Server Real Time	Server CPU Time
Traditional Merge	33 hours	64 minutes	11 hours	57 minutes
SAS Writes WHERE	74 minutes	10 minutes	39 minutes	9 minutes
KEY=	19 minutes	16 seconds	9 minutes	11 seconds
SQL Join with Index	19 minutes	20 seconds	8 minutes	17 seconds
SQL Join, No Index			43 minutes	3 minutes

Other Resources:

Table Lookup: Techniques beyond the Obvious

Nancy Cronin, CC Training Services, Belgium
ir. Henri Theuwissen, SOLID Partners, Belgium
<http://www2.sas.com/proceedings/sugi27/p011-27.pdf>

Using the Magical Keyword "INTO:" in PROC SQL

Thiru Satchi
Blue Cross and Blue Shield of Massachusetts, Boston,
Massachusetts
<http://www2.sas.com/proceedings/sugi27/p071-27.pdf>

Table Look-up Using Techniques Other Than the Matched Merge DATA Step

Sandra Lynn Aker, Alysandra Lynn, Inc., Bensenville, IL
<http://www2.sas.com/proceedings/sugi27/p195-27.pdf>