

PhUSE 2007

Paper CC01

SAS Indexes – Benefits, limits and applications

Carole Beaugendre, Quintiles, Strasbourg, France

ABSTRACT

SAS® indexes are not widely used although they may considerably improve performance in specific cases. It is useful for programmers to know about SAS indexes and how to use them. However they should be used with caution as a badly designed index or an index used inappropriately may be less efficient than using no index. An interesting application is the creation of patient profiles, especially in large clinical studies.

INTRODUCTION

An index, like any tool, is more beneficial when understood. Although this paper briefly describes how to create and use indexes, the main question discussed is the following: When is it appropriate to use an index? A formal and exhaustive description of SAS code and numerous options is not within the scope of this paper. An interesting application of indexes is also described at the end of the paper: the creation of patient profiles, especially in large clinical studies where the run-time can be long, creating a burden on computing resources.

DEFINITION OF SAS INDEXES

An index is an optional file that can be created for a SAS data set to provide direct access to specific observations. The values are stored in ascending order for a given variable (or variables) and information as to the location of those values within the data set is recorded.

Without an index, SAS accesses observations sequentially (reading all observations until the observation is found or all observations are read) whereas with an index, SAS accesses the requested observation(s) directly.

For each data file, one or more index variables can be created (using different key variables). However, there is only one index file per data file: all index variables relating to the same data file are stored in a single file. The index file consists of entries that are organized hierarchically and connected by pointers. Each entry consists of a distinct value (thereafter called “key value” or “key variable”) and one or more unique record identifiers (RID) that identify each observation of the source data set containing the key value.

For instance, in the table below, an index was created using the patient number as key value. You can think of the source data file as a set of measurements taken for each patient at different visits during the study. The objective here is to extract data for one specific patient. When an index is used to process a request, such as a WHERE expression, SAS looks for the requested key value (e.g. patient number 02060001) and uses the RID to read the corresponding observations in the source data file (e.g. RID 6, 14 and 78). The unique record identifiers should actually be regarded as internal observation numbers.

In a data set with thousands of patients, using the patient number as an index variable might help to locate patient’s observations more efficiently.

Patient number (key variable)	Unique record identifier (RID)
02060001	6, 14, 78
02060002	12, 54
02060005	37

PhUSE 2007

HOW TO USE SAS INDEXES

SIMPLE INDEX/COMPOSITE INDEX

An index can be simple (built from a single key variable) or composite (built from multiple variables that constitute a combined key). In all cases, the key should be carefully chosen according to how observations are to be selected within the data sets.

As already seen above, in a clinical study, the patient number is a good candidate for the creation of a simple index. Indeed, this number is unique for each patient included in the study. However, if you need to extract e.g. baseline data only, it may be interesting to create a composite index using both patient number and visit number. In this case, a fewer number of observations will be extracted from the data file than with a simple index built from the patient number.

When you are deciding whether to create a simple index or a composite index, consider how you will access the data. If you often access data for a single variable, a simple index will do. But if you frequently access data for multiple variables, a composite index could be more beneficial.

In any case, the more discriminating the index chosen, the fewer observations returned, and the better the performance of the selection process.

CREATION OF INDEX

An index can be created either at the time a data set is created or from an existing data set. Once an index exists, SAS treats it as part of the data set. That is, if values are modified or observations added or deleted, the index is automatically updated.

There are two main SAS procedures that enable you to create an index: PROC DATASETS or PROC SQL.

The first example below shows the use of the DATASETS procedure for the creation of two indexes from an existing data set (source_data): one simple index using the patient number (patnum) as a key variable, and one composite index using the combination of patient number (patnum) and visit number (visnum) as a new key variable named patvis. Then the second example shows the same creation of two indexes from an existing data set but using the SQL procedure.

An index can also be created in a DATA step using the INDEX= option as shown in example 3.

Example 1:

```
proc datasets;
  modify source_data;
    index create patnum;
    index create patvis=(patnum, visnum);
run;
```

Example 2:

```
proc sql;
  create index patnum on source_data (patnum);
  create index patvis on source_data (patnum, visnum);
quit;
```

Example 3:

```
data source_data (index=(patnum));
data source_data (index=(patvis=(patnum, visnum)));
```

There is not one method better than the others. The choice is rather a function of the programmer's preferences and own habits. Examples 1 and 2 show how to create indexes from an existing data set. Example 3 shows how to create an index at the time a data set is created.

USE OF INDEXES

Once created, indexes may be used in one of the following situations: to select observations using a WHERE or BY statement in a DATA or PROC step (see example 4 below), or the KEY option on a SET or MODIFY statement (see example 5).

PhUSE 2007

Example 4: selection of a subset of observations

```
data select_data;
  set source_data;
  where patnum='02060002';
run;
```

Example 5: modification of observations located by an index:

In this example, the source data set (indexed on patnum) is modified using new information from the data file patient_desc. The KEY option tells SAS that an index exists and may be used to locate the observations to be modified.

```
data source_data;
  set patient_desc;
  modify source_data key=patnum;
  nbvisits=nbvisits+new_visits;
run;
```

In all structures other than those stated above, even if an index exists, it will never be used. It should also be noted that an index will never be used to sort data (BY in PROC SORT) either. In any of the structures stated above, it should also be noted that an index is not automatically used. SAS will first decide whether using an existing index is more efficient than not using it. A lot of factors are taken into account, such as the size of the data set, which index is available, or the number of observations to be returned from the selection process. SAS also uses the “centiles” information.

Centiles is the SAS key word for “cumulative percentiles”. There are 21 centiles values stored in the index descriptor: 0, 5, 10, 15, 20... 100 (i.e. from 0 to 100 by steps of 5). They represent the percentiles of key variables of the index. For instance, the 5th centile contains the key value such that 5 % of observations have a key value lower than this flagged value. Thus, 0 contains the lowest key value, whereas 100 contains the highest. Centiles are a good indicator of the added value of using an index in any situation.

For instance, if the age of the patient is used as a key variable and a query asks for patients younger than 50, centiles will give the information of the number of observations to return and SAS will be able to decide whether using the existing index will be more efficient or not. If the tenth centile equals 52, this means that less than 10 % of the data match the query.

In some cases, SAS does not even bother comparing resource usages and automatically uses the existing index. This is the case when SAS estimates that the number of qualified observations is less than 3 % of the data file. SAS often takes the right decision regarding whether to use an existing index or not. To know if an index has been used, there is an option that can be added to print the information in the log window:

```
OPTION MSGLEVEL=I;
```

Using this option with example 4 above, the message posted will be the following if the index was actually used:

```
INFO: Index PATNUM selected for WHERE clause optimization.
```

To override SAS's decision, the following options can be specified in the SET statement within a DATA step:

- IDXNAME= to force which index should be used (if any index is used).
- IDXWHERE="Yes" to tell SAS to choose the best index to optimize a WHERE expression, and to disregard the possibility that a sequential search of the data set might be more resource-efficient / "No" to tell SAS to ignore all indexes and satisfy the conditions of a WHERE expression with a sequential search of the data set. IDXWHERE= cannot be used to override the use of an index to process a BY statement.

These two options are mutually exclusive. If you issue the system option MSGLEVEL=I, you can request that IDXNAME= or IDXWHERE= usage is noted in the SAS log if the setting affects index processing.

BENEFITS AND LIMITS

Even though an index can reduce the time required to locate a set of observations, especially for a large data set, there are costs associated with creating, storing and maintaining the index.

PhUSE 2007

Basically, indexes can dramatically improve the performance of programs that frequently access small subsets of observations from large data sets. These are the two main considerations for deciding whether to create an index or not. Indeed, index files need computer resources to be created and maintained up-to-date when the corresponding data set is modified. Thus the more frequently they are used, the more cost effective they are. Moreover, if a large subset of observations is selected, a sequential reading may be as efficient in terms of computer resources as creating and using an index.

The table below gives some guidelines. However, many factors in an operating environment can affect performance.

Subset size	Indexing action
1%-15%	Program performance definitely improved
16%-20%	Program performance probably improved
21%-33%	Program performance possibly improved or worsened
>33%	Program performance not improved. A sequential read of the entire data set is probably more efficient.

Basically, if more than 20% of observations are to be extracted from the data set, it does not make sense to create an index. Resources would be used for the creation of the index, and the selection process would not be improved. This would lead to no better, or even worse, program performance.

Therefore, it is important to carefully choose when it is appropriate to use an index. When deciding whether to create an index, you must consider increased resource usage together with the performance improvement.

APPLICATIONS

WHAT ARE PATIENT PROFILES?

In many clinical studies, specific data reports called "patient profiles" are used for medical review purposes. Indeed, they allow medical reviewers to check all the data of each patient throughout the study and thus to detect any abnormal data. Sometimes these involve only a small selection of the patient's data, but often all CRF data are included. In this case, there are as many reports as patients in the study and each single data set is accessed once for each single patient. Consequently, the running process can be very burdensome, especially in large clinical studies where thousands of patients are enrolled.

Below is an example of a patient profile listing for study drug administration. Each data of the CRF page is reported at each visit for this particular patient.

Protocol: xxxx		Center 040 ,		Patient No. 9000			

(DSMB data report - clinical data cut off : 22JUL2007)							
Subjects individual safety profile 9: Study drug administration							
	Drug	If yes, date	If no, reason	Start	Stop	Dose level	Total dose
Visit	adm.?	of infusion		time	time	(mg/m2)	(mg/m2)

V001	Yes	07APR2007		16:40	18:37	400.00	715.00
V002	Yes	15APR2007		ND:ND	ND:ND	250.00	435.00
V003	No		Toxicity				
V004	Yes	28APR2007		17:00	17:55	250.00	432.60
V005	Yes	06MAY2007		16:55	17:40	250.00	432.60
V006	Yes	12MAY2007		17:05	18:02	250.00	444.00
V007	Yes	20MAY2007		11:41	12:36	250.00	438.00
V008	Yes	26MAY2007		16:47	17:48	250.00	441.00

WHY IS THIS CASE APPROPRIATE?

As already seen, there are two main conditions for the creation of index to be appropriate: the frequency of use and the extraction of a small subset from a large data set. Both these conditions are filled by patient profiles. Indeed, the index will be used as many times as the number of patients in the study, and at each time, data relating to one patient form a small subset of the whole database.

PhUSE 2007

Moreover, as patient profiles are most of the time based on CRF data, there is no modification to be expected in the source data files and indexes do not need to be changed. However, indexes need to be updated at each new database extract.

INDEXES TESTED ON REAL DATA

I have encountered a “perfect” case for testing the performance of the index tool and comparing performance of the running process with and without using indexes. This case was a large phase III study in oncology, with 2082 patients enrolled for several years, and 47 CRF data files. The patient profiles were to be sent each quarter for safety monitoring purposes. Each time, $2082 * 47 = 97854$ accesses to the data sets were necessary. The table below gives some statistics regarding process times according to the use of the index tool.

	Using indexes	Using no index
Average time needed to run one patient profile (access to 47 data files)	16 sec	2 min
Average total time needed to run 2082 patient profiles*	9 hours and 15 min	More than 69 hours
Additional time (creation of index)	4 min 57 sec	--
Total time	9 hours and 20 min	More than 69 hours

* assuming that the running process takes the same time for all patients, which is not true since it is function of the amount of data relating to each patient. However, this is a good estimation of the time needed for the whole process.

This case is obviously a “perfect” case essentially due to the size of the database (number of patients and amount of data for each patient). This shows that the index can be a very powerful performance tool in such cases. The demonstration is not so striking when the database is smaller, but from my experience, using indexes for the production of patient profiles is always beneficial.

CONCLUSION

Indexing can be a powerful performance tool for certain situations, where a small subset of observations is to be selected from a large data set. However, even though an index can reduce the time required to locate a set of observations, there are costs associated with creating, storing and maintaining the index.

The main considerations are the size of the source data set, the size of the subset, the frequency of use and also the frequency of source data modifications.

An appropriate application is the production of patient profiles, especially in large clinical studies. Indeed, this case fulfills all the conditions required for this tool to be performant: the index will be used as many times as the number of patients in the study, and each time, data relating to one patient makes up a small subset of the whole database. Even in smaller studies, indexes remain performant for the creation of patient profiles.

REFERENCES

Michael A. Raithel. 2004. *Creating and exploiting SAS Indexes*. SUGI 29, paper 123-29.

Michael A. Raithel. 2005. *The basic of using SAS Indexes*. SUGI 30, paper 247-30.

SAS OnlineDoc®, Version 9, Copyright 2007, SAS Institute, Inc.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Carole Beaugendre

Quintiles

Email: carole.beaugendre@quintiles.com

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.

Other brand and product names are trademarks of their respective companies.