Paper 095-2008

Using Table Lookup Techniques Efficiently

Jane Stroupe & Linda Jolley, SAS Institute Inc., Cary, NC

ABSTRACT

With the volume of data that exists in a data warehousing environment, it is often necessary to combine data in order to create a meaningful report. Many techniques offer this type of table lookup: arrays, hash objects, formats, joins, merges, indexes, or a combination of any of these. With so many ways to perform a table lookup, how do you decide which one to use? This paper investigates the how and why of each of these techniques in order to give you a better idea about where to begin.

INTRODUCTION

Lookup values for a table lookup can be stored in the following ways in SAS:

- code
- array
- hash object
- format
- data set

The following techniques can be used to perform table lookups:

- IF/THEN or SELECT/WHEN statements
- array index value
- hash object key value
- FORMAT statement, PUT function
- merge, join, KEY= option

The data sets that are used in this tutorial were created using the following programs:

DETAIL TABLE: MYLIST

```
data MyList;
   input Date:date9. Code:$3. AreaCode $ PhoneNum $ ToAcronym:$5.
FromAcronym:$5.;
   format date date9.;
   datalines;
21MAR2008 101 312 555-6674 OTOH LOL
21MAR2008 103 312 555-5098 JTLYK OIC
21MAR2008 106 312 555-9088 AFAIK DQMOT
21MAR2008 101 312 555-6674 OIC IAC
21MAR2008 103 312 555-5098 PM GMTA
21MAR2008 103 312 555-5098 BTDT IHU
21MAR2008 101 312 555-6674 BEG TU
21MAR2008 101 312 555-6674 BFN SYS
22MAR2008 106 312 555-9088 C&G AFAIK
22MAR2008 106 312 555-9088 MLO TPTB
22MAR2008 102 212 555-0038 GMTA YUP
22MAR2008 108 212 555-1174 HTH IHU
23MAR2008 105 212 555-3452 GA DQMOT
```

```
23MAR2008 105 212 555-3452 IOW YBS
23MAR2008 104 212 555-4544 NP IMS
23MAR2008 104 212 555-4544 WU? WFM
23MAR2008 102 212 555-7456 JTLYK TAFN
23MAR2008 102 212 555-7456 ROTFL BBL
;
run;
```

LOOKUP TABLE: PHONEBOOK

```
data PhoneBook;
   infile datalines dsd dlm=' ';
   input Name:$14. PhoneNum:$8. SecretCode:$3.;
datalines;
"Superman" 555-6674 101
"Spider-Man" 555-5098 102
Batman 555-9088 103
"Wonder Woman" 555-3452 104
Hulk 555-7483 105
Catwoman 555-8943 106
"Captain Marvel" 555-4544 107
Joker 555-4755 108
;
run;
```

LOOKUP TABLE: ACRONYM

```
data Acronyms;
  infile datalines dsd;
  input Acronym:$5. Meaning:$50. ;
  datalines;
AFAIK,as far as I know
BBL,be back later
BEG,big evil grin
BFN,bye for now
BTDT,been there done that
DQMOT,don't quote me on this
GA,go ahead
GMTA,great minds think alike
HTH,hope this helps
IAC,in any case
IHU,I hear you
IMS,I am sorry
IOW,in other words
JTLYK,just to let you know
LOL,laughing out loud
NP,no problem
OIC,"oh, I see"
OTOH,on the other hand
PM,private message
ROTFL,rolling on the floor laughing
SYS,see you soon
TAFN,that's all for now
```

```
TPTB,the powers that be
TU,thank you
WFM,works for me
WU?,what's up?
YBS,you'll be sorry
;
run;
```

USING IF/THEN OR SELECT/WHEN STATEMENTS

PROGRAM

```
data location;
   set mylist;
   if AreaCode='312' then Location='Chicago, IL';
   else if AreaCode='212' then Location='New York, NY';
   else Location='Unknown';
run;

data location;
   set mylist;
   select (AreaCode);
     when('312') Location='Chicago, IL';
     when('714') Location='New York, NY';
     otherwise Location='Unknown';
   end;
run;
```

GUIDELINES

Use IF-THEN/ELSE statements in the following circumstances:

- You have only a few conditions to check.
- The data values are not uniformly distributed.
- The values are character or discrete numeric data.
- There are bounded ranges of data (for example, 1<x<2).
- For mutually exclusive conditions, use the ELSE-IF statement instead of an IF statement for all conditions, except the first.
- Check the most frequently occurring condition first.
- When you execute multiple statements based on a condition, put the statements into a DO group.

Use a SELECT statement when you have a long series of mutually exclusive conditions.

USING ARRAYS

PROGRAM

```
data secret;
   keep SecretPal AreaCode PhoneNum;
   if _n_=1 then do i=1 to numobs;
       set PhoneBook nobs=numobs;
       array names{101:108} $ 18 _temporary_ ;
       names{SecretCode}=Name;
   end;
   set mylist;
   SecretPal=names{Code};
run;
```

ADVANTAGES

Using an array offers the following advantages:

- Positional order can be used.
- Multiple values can determine the array element to be returned.
- You can use a non-sorted and non-indexed base data set.
- Numeric mathematical expressions can determine which element of the array to be looked up; an exact match is not required.
- An array will be faster than a hash object or format lookup.

DISADVANTAGES

Using an array has the following disadvantages:

- A large amount of memory may be required to load the entire array.
- You must have a numeric value as pointer to the array elements.
- Arrays can be used only in the DATA step.
- An array can return of only a single value from the lookup operation.
- The dimensions must be supplied at compile time by either hardcoding or macro variables.

USING THE HASH DATA STEP OBJECT

PROGRAM

```
data Messages;
   keep ToAcronym FromAcronym ToMeaning FromMeaning PhoneNum;
   length Acronym $5 Meaning $50;
   if _n_=1 then do;
      declare hash wh(dataset:'Acronyms');
      wh.definekey('Acronym');
      wh.definedata('Meaning');
      wh.definedone();
      call missing(Acronym, Meaning);
   end;
   set mylist;
   rc=wh.find(key:ToAcronym);
   if rc=0 then ToMeaning=Meaning;
   else ToMeaning='Unknown';
   rc=wh.find(key:FromAcronym);
   if rc=0 then FromMeaning=Meaning;
   else FromMeaning='Unknown';

run;
```

ADVANTAGES

Using hash objects offers the following advantages:

- The key values can be either character and numeric.
- Composite keys can be used.
- Table lookup is faster than formats, merges or joins.
- The hash object can be loaded from a SAS data set.
- You have a fine level of control (flexibility).
- You can do chained lookups.

DISADVANTAGES

Using hash objects has the following disadvantages:

- A unique key is required. (Duplicate keys are allowed in SAS 9.2.)
- Hash objects can be used only in the DATA step.
- A large amount of memory may be required to load the entire hash object.

USING A FORMAT

PROGRAM

```
data fmt;
   set acronyms(rename =(acronym = start meaning = label));
   retain fmtname '$Acronym';
run;

proc format cntlin=fmt library=sasuser;
   value $Extra 'C&G','MLO','YUP' = 'Unknown'
                         other = [$acronym.];
run;

options fmtsearch =(sasuser);

data combine;
   keep ToAcronym FromAcronym ToMeaning FromMeaning
        PhoneNum;
   set mylist;
   ToMeaning = put(ToAcronym, $Extra.);
   FromMeaning = put(FromAcronym, $Extra.);
run;
```

ADVANTAGES

Using formats offers the following advantages:

- Format are familiar to most SAS programmers.
- There is no need to create additional data.
- Formats can be used with procedures.
- Range searches can be performed for both character and numeric values.
- Formats use a binary search through lookup table for retrieving values.
- Formats can be used to centralize maintenance of the lookup table.
- You can use multiple PUT functions to create multiple variables.

DISADVANTAGES

Using formats has the following disadvantages:

- A large amount of memory may be required to load the entire the entire format for the binary search.
- Only one variable is used for the table lookup.
- More disk space is required to store a format than to store the equivalent SAS data set.

USING THE DATA STEP MERGE

PROGRAM

```
proc sort data = acronyms;
   by Acronym;
run;

proc sort data = mylist;
   by ToAcronym;
run;

data ToMsg;
   keep ToAcronym FromAcronym ToMeaning PhoneNum;
   merge acronyms(in = a rename =(Acronym = ToAcronym))
         mylist(in = m);
   by ToAcronym;
   if m;
   if a then ToMeaning = Meaning;
   else ToMeaning = 'Unknown';
run;

proc sort data = ToMsg;
   by FromAcronym;
run;

data BothMsg;
   keep ToAcronym FromAcronym ToMeaning FromMeaning
        PhoneNum;
   merge acronyms(in=a rename =(Acronym = FromAcronym))
         ToMsg(in = t);
   by FromAcronym;
   if t;
   if a then FromMeaning = Meaning;
   else FromMeaning = 'Unknown';
run;
```

ADVANTAGES

Merging offers the following advantages:

- Multiple values can be returned.
- There is no limit to the size of the table, other than disk space.
- Multiple BY variables enable lookups that depend on more than one variable.
- Multiple data sets can be used to provide access to different tables.
- A merge enables complex business logic to be incorporated into the new data set by using DATA step processing, such as arrays and DO loops, in addition to merging features.

- The IN= data set option and subsequent IF-THEN/ELSE logic afford comprehensive control over whether to accept, reject, or process differently, depending on which data set contributed each observation.
- Observations with duplicate BY values are joined one-to-one instead of being expanded into a Cartesian product, as SQL does.

DISADVANTAGES

Merging has the following disadvantages:

- Data sets must be sorted by or indexed based on the BY variable(s).
- An exact match on the key value(s) must be found.
- The BY variable(s) must be present in all data sets.
- When more than one data set contributes variables with the same name, the values from the variables in the rightmost data set overwrite the other like-named variables, and no warning is printed.

USING THE SQL INNER JOIN

PROGRAM

```
proc sql;
   create table SQLBothMsg as
      select ToAcronym,
             FromAcronym,
             a1.Meaning as ToMeaning,
             a2.Meaning as FromMeaning,
             PhoneNum
           from Mylist,
               Acronyms as a1,
               Acronyms as a2
               where a1.Acronym = ToAcronym and
                   a2.Acronym = FromAcronym;
quit;
```

ADVANTAGES

Using the SQL inner join offers the following advantages:

- Multiple data sets can be joined without having common variables in all data sets.
- Data sets do not have to be sorted or indexed.
- You can perform inequality joins.
- You can create data files (tables), views, or reports.
- PROC SQL follows ANSI standard language definitions, so that you can use knowledge gained from other languages.
- Duplicate BY values are combined into a Cartesian product.

DISADVANTAGES

Using the SQL Inner Join has the following disadvantages:

- The maximum number of tables that can be joined at one time is 256, beginning in SAS 9.1.3 Service Pack 4.
- PROC SQL might require more resources than the DATA step MERGE statement for simple joins.
- Complex business logic is difficult to incorporate into the join.
- Duplicate BY values are combined into a Cartesian product, which can produce an extremely large output data set.

THE DATA STEP MERGE STATEMENT OR THE SQL PROCEDURE?

Whether the DATA Step MERGE or the PROC SQL statement is more efficient depends on your data and these factors:

- the relationship between the tables
- the sparseness or denseness of matches
- the size of the tables
- the availability of an index or sort flag

Use these general guidelines to help you determine whether to use the DATA Step MERGE or PROC SQL statement:

- When data sets are large and unsorted, the SQL inner join might outperform SORT and MERGE.
- If you have a long series of SORT and DATA steps, the SQL inner join might be easier to code and comprehend.
- In most cases, a DATA step MERGE statement outperforms an SQL outer join, even taking sort resources into account.
- One exception is a very sparse match join when you only want the observations with matching key values.
- Keep in mind that the SQL procedure and the DATA step MERGE do not provide the same results if you have a many-to-many match.

Because there are no hard and fast rules about the efficiency of MERGE vs. SQL, you will have to benchmark your SAS code to see how it performs with your data. Remember that benchmarking involves multiple submissions of the two sets of code against the data.

USING THE SET/SET KEY= STATEMENTS

PROGRAM

```
proc datasets library = work;
   modify Acronyms;
   index create Acronym/unique;
run;

data KEYMsg1(rename =(Acronym = ToAcronym));
   keep Acronym FromAcronym ToMeaning PhoneNum;
   set mylist(rename =(ToAcronym = Acronym));
   set Acronyms key = Acronym;
   if _iorc_=0 then ToMeaning = Meaning;
   else ToMeaning = 'Unknown';
run;

data KEYMsgs(rename =(Acronym = FromAcronym));
   keep ToAcronym Acronym ToMeaning FromMeaning PhoneNum;
   set KEYMsg1(rename =(FromAcronym = Acronym));
   set Acronyms key = Acronym;
   if _iorc_ = 0 then FromMeaning = Meaning;
   else FromMeaning = 'Unknown';
run;
```

ADVANTAGES

Using the DATA step SET/SET KEY= offers the following advantages:

- Only the necessary observations are read.
- An existing index is used.
- Multiple values can be returned.
- Availability of DATA step syntax provides the full power of the DATA step.
- Exact matches are returned.
- The automatic _IORC_ variable can be used to control non-matching data.

DISADVANTAGES

The DATA step SET/SET KEY= has the following disadvantages:

- An index on one data set is required.
- Creating and maintaining an index uses resources.
- It is useful only for data with exact matches.
- When the indexed data set is not sorted by the key variable(s), the I/O could increase considerably because of the random access of the master data set and the I/O required to access the index.

CONCLUSION

IN MEMORY METHODS

	Array	Hash Object	Format
Where is it used?	DATA step only.	DATA step only.	DATA steps and procedures.
Is it permanently stored?	No; you have to repeat the code every time you use the lookup table.	No; you have to repeat the code every time you use the lookup table.	Can be stored permanently so the code to create the lookup table has to be run only one time.
When is it loaded into memory?	When array is created.	When hash object is created.	When the format is used.
Can you load it from a SAS data set?	Yes.	Yes.	Yes.

OTHER METHODS

	IF/ THEN	MERGE	SQL	SET/SET & KEY=
Processing Method	Sequential	Sequential	Cartesian product	Direct access using index
Memory Used	Data set page(s)	Data set page(s)	Utility work space to form Cartesian product	Data set page(s) including all page(s) for index

CONTACT INFORMATION

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

Jane Stroupe
SAS Institute Inc.
Chicago Regional Office
Work Phone: (847) 367-7216
E-mail: Jane.Stroupe@sas.com

Linda Jolley
SAS Institute Inc.
Kansas City Regional Office
Work Phone (913) 491-1166
E-mail: Linda.Jolley@sas.com