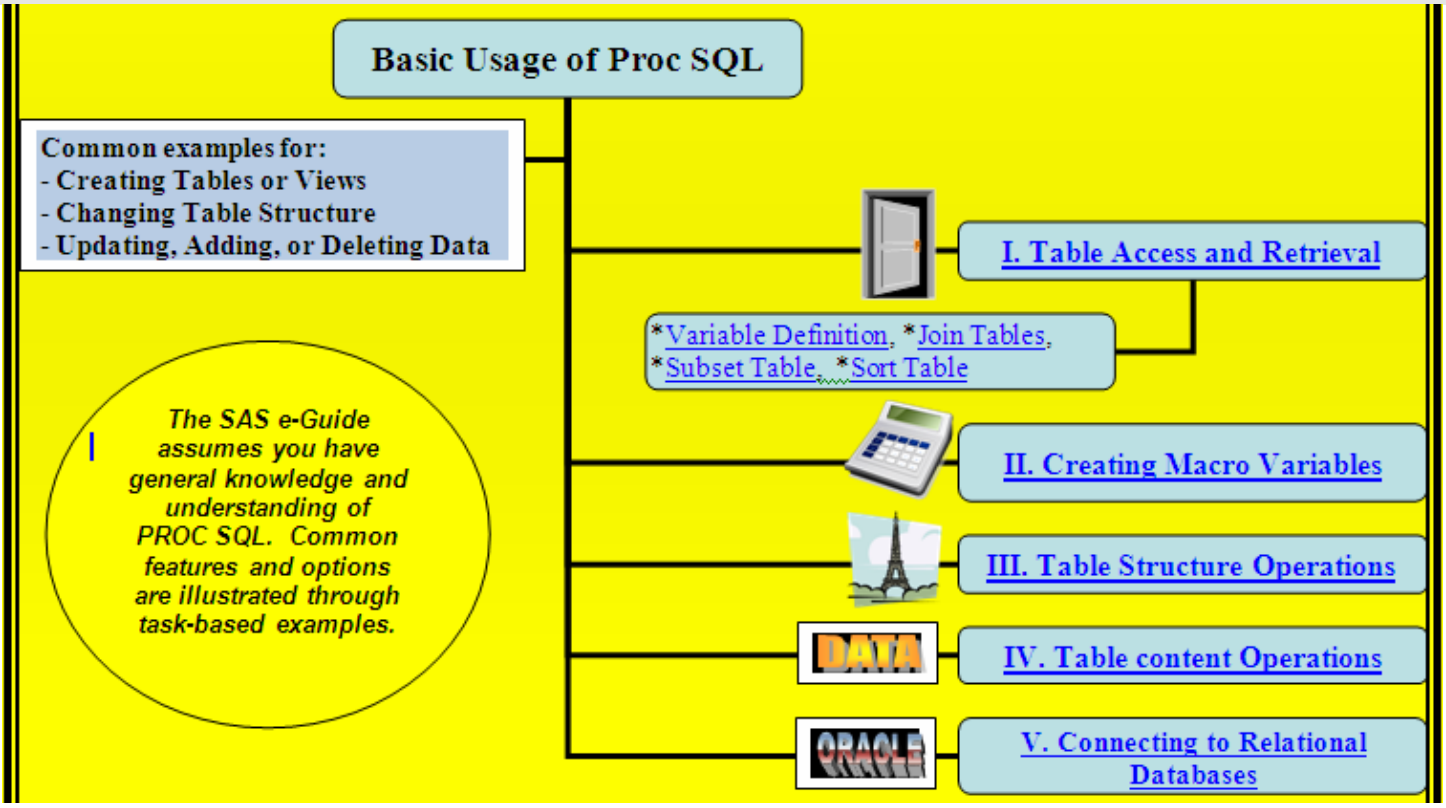
Quick Results with Proc SQL – *New SAS e-Guide*

(Ready to increase *your productivity* using the single most powerful SAS Procedure?)

Through task-based examples, experienced SAS users can read-on-purpose to quickly apply the correct SAS syntax in their SAS program for increased productivity. Multiple programming options and hyperlinks to relevant SAS books and technical papers help to better reinforce and understand essential programming concepts and syntax.

1. **The single most concise productivity e-Guide available.** In general, 80% of Proc SQL's productivity comes from 20% of Proc SQL's syntax. Below is the Quick Results with Proc SQL e-guide outline.



2. **Task-based examples to show correct SAS syntax.** Task-based examples include table access and retrieval, creating macro variables, table structure operations, table content operations and connecting to relational databases.

Basic Lines of Code for Table Access and Retrieval:

```
1. create table mytable as
2. select name, sex
3. from sashelp.class
4. where sex = 'F'                /* optional */
5. order by name                  /* optional */
6. ;
```

Line 1: to create table mytable.
Line 2: select variables. See a. Variable Definition.
Line 3: from source table. See b. Join Tables.
Line 4: apply condition for subsetting table. See c. Subset Table.
Line 5: sort result table. See d. Sort Table.

3. **Multiple programming options in one central place.** This unique approach empowers programmers to think beyond the box for maximum understanding. Examples of programming

http://www.sascommunity.org/wiki/Sunil_Gupta#Sunil.27s_Top_10_PROC_SQL_Papers

options include five options for creating variables, four options for selecting variables, and five options for creating macro variables.

Five Options for Creating Variables: Select Clause

```
1. select int((age+150)/10) as ■yage
2. select ■ax(height, weight) as ■axval
3. select ((weight/su■(weight))*100) as wpercnt
4. select 'my constant' as ■yna■e
5. select case
              when age  < 13                          then 1
              when age  between 13 and 15  then 2
              when age > 15                           then 3
              else                                    .
   end as depthead
```

Five Options for Creating Macro Variables:

```
1. select presult into :fnpv fro■ ■ylib.statresult;

2. select presult into :fnpv separated by ',' fro■ ■ylib.statresult;

3. select left(put(count(patient), 3.))
   into :fnpv1 - :fn5
   fro■ ■ylib.statresult where tx ne . and study = 94123
   group by tx;

4. select avg(salary), ■in(salary), ■ax(salary) into
  :■ean :■in :■ax  fro■ ■ylib.hrstaff;

5. select case
      when round(< p_c■hr■s >, 0.001) < 0.05 then
      (put(< p_c■hr■s >, 5.3) || '*')
      else put(< p_c■hr■s >, 5.3)
   end into :fnpv fro■ ■ylib.statresult;
```

4. **Key SAS books and references for more engagement.** SAS users can link to relevant SAS books, SAS Webcasts and Podcasts as well as popular SAS technical papers to get more involved while accessing more detail information.

Key SAS books and references for more information

PROC SQL: beyond the basics using SAS
The Essential PROC SQL Handbook for SAS Users
PROC SQL by Example: Using SQL within SAS

Webcast: Proc SQL Tips and Techniques, Dictionary Tables

Sunil's top 10 recommended Proc SQL Papers – *Get involved*, e-mail me to consider your favorite SAS paper or SAS reference

http://www.sascommunity.org/wiki/Sunil_Gupta#Sunil.27s_Top_10_PROC_SQL_Papers