

Summary Sheet: Ready To Become Really Productive Using PROC SQL?

Sunil K. Gupta, http://www.sascommunity.org/wiki/Sunil_Gupta

```
proc sql;                               /* PROC SQL Basic Example Usage */
  select name, sex
  from sashelp.class
  where sex = 'F'
  < group by > < having >                < Additional Options >
  order by name; quit;
```

```
PROC SQL;                               /* Anatomy of PROC SQL General Usage */
CREATE table mytable as
/* Nine Benefits: Validate/Create/Drop Views or Tables,
   Create/Alter/Update/Insert/Delete Variables */
/* 1. Four Main Components: a. SELECT, b. FROM, c. WHERE, d. ORDER */
A. SELECT name, sex

/* 2. Four selecting columns options:
  a. ',' to separate columns
  b. label=' ' format= $10. length=10 to add attributes
  c. '*' to select all columns
  d. distinct to select unique columns */

/* 3. Eight creating columns options:
  a. functions ex. init((age + 150)/10) as myage
  b. summary function ex. max(height, weight) as maxval
  c. summary function ex. ((weight/sum(weight))*100) as wpercent
  d. constant ex. 'my home' as myhome
  e. character expression ex. city || ',' || state as address
  f. select case when age < 13 then 1 else 0 end as agegrp */
  g. select case age when < 13 then 1 else 0 end as agegrp */
  h. select sum(sex='M') as nmale, sum(sex='F') as nfemale

/* 6. Five macro variable creating options:
  a. into : to store one value in one macro variable
  b. into : separated by to store multiple values
  c. into : - : to create multiple macro variables
  d. summary function into: to create one macro variable
  e. select-case into: to create one macro variable */

B. FROM sashelp.class as class,
   Mylib.students as students
/* Four main join options: inner matching/outer LEFT/FULL/RIGHT JOIN */
/* FROM <DS1> <FULL JOIN> <DS2> ON <DS1.VAR1> = <DS2.VAR2> */
```

C. WHERE class.name = students.name and class.sex = 'F'

/* 4. Four subsetting options:

- a. direct variable using where clause
- b. calculated variable using where clause
- c. function, ex. index(name, 'B') using where clause
- d. summary function, ex. sum(sales) > 0 using having clause */

/* 5. Two main subquery options:

- a. one value returned
- b. multiple values returned with <Variable> < IN Operator>
(SELECT <Variable> FROM <Table> WHERE <Condition Expression>) */

D. ORDER by name

/* Two sorting options: order/group by calculated, desc */
; QUIT;

SASHELP.CLASS SAMPLE DATA SET

Obs	Name	Sex	Age	Height	Weight
1	Alice	F	13	56.5	84.0
2	Barbara	F	13	65.3	98.0
3	Carol	F	14	62.8	102.5
4	Jane	F	12	59.8	84.5
5	Janet	F	15	62.5	112.5
6	Joyce	F	11	51.3	50.5
7	Judy	F	14	64.3	90.0
8	Louise	F	12	56.3	77.0
9	Mary	F	15	66.5	112.0
10	Alfred	M	14	69.0	112.5
11	Henry	M	14	63.5	102.5
12	James	M	12	57.3	83.0
13	Jeffrey	M	13	62.5	84.0
14	John	M	12	59.0	99.5
15	Philip	M	16	72.0	150.0
16	Robert	M	12	64.8	128.0
17	Ronald	M	15	67.0	133.0
18	Thomas	M	11	57.5	85.0
19	William	M	15	66.5	112.0