



# Back to BASE - PROC SQL

Mike Olson

TCASUG

12/17/09



# PROC SQL – What is it?

- Structured Query Language
  - SQL
    - “Sequel” to data folks
    - “Ess – que – ell” to everyone else
  - Not a SAS specific
    - Your SQL skills are transferrable to other systems!
      - Netezza
      - Access
      - MySQL
      - SQL Server
      - Many others!



# PROC SQL – What can it do?

- Retrieve data
  - Data can be from SAS data sets, external databases, or both
- Join multiple tables
  - Columns names need not be the same
  - Data does not need to be sorted
- Add or modify data values
- Add, modify, or drop columns
- And much more...



# Presentation Overview

- SAS – PROC SQL Dialect
- Beginning Queries
- SQL Functions
- Advanced Queries
- Combining Tables
- Macro Variables

# PROC SQL – SAS Dialect

Data Processing	SAS	SQL
file	SAS data set	table
record	observation	row
file	variable	column

- Statements and Clauses

proc sql; ← Statements end with semicolon  
select name, age, gender, height, weight  
weight / height as ratio  
from sasuser.heightsweights  
where age < 45  
order by height; ←

SQL  
Clauses

RUN; Statement not needed

# The Santa Clause



# PROC SQL – Syntax

```
PROC SQL options;  
  SELECT column(s)  
  FROM table-name | view-  
  name  
  WHERE expression  
  GROUP BY column(s)  
  HAVING expression  
  ORDER BY column(s);  
QUIT;
```

invokes the SQL procedure

specifies the column(s) to be selected

specifies the table(s) to be queried

conditionally subsets the data

classifies the data into groups based on the specified column(s)

uses an expression to subset or restrict groups of data based on a group condition

sorts the rows that the query returns by the value(s) of the specified column(s)

# Beginning Queries

## Selecting all columns:

```
proc sql;  
select * from  
sashelp.prdsale;  
quit;
```

This statement is similar to:  
proc print data = sashelp.prdsale; run;

The asterisk (\*) selects all columns

Actual Sales	Predicted Sales	Country	Region	Product Division	type	Product	Quarter	Year	Month
\$925.00	\$850.00	CANADA	EAST	EDUCATION	FURNITURE	SOFA	1	1993	Jan
\$999.00	\$297.00	CANADA	EAST	EDUCATION	FURNITURE	SOFA	1	1993	Feb
\$608.00	\$846.00	CANADA	EAST	EDUCATION	FURNITURE	SOFA	1	1993	Mar
\$642.00	\$533.00	CANADA	EAST	EDUCATION	FURNITURE	SOFA	2	1993	Apr
\$656.00	\$646.00	CANADA	EAST	EDUCATION	FURNITURE	SOFA	2	1993	May
\$948.00	\$486.00	CANADA	EAST	EDUCATION	FURNITURE	SOFA	2	1993	Jun
\$612.00	\$717.00	CANADA	EAST	EDUCATION	FURNITURE	SOFA	3	1993	Jul
\$114.00	\$564.00	CANADA	EAST	EDUCATION	FURNITURE	SOFA	3	1993	Aug

# Beginning Queries (cont.)

## Selecting specific columns:

```
proc sql;  
  select country, actual  
  from sashelp.prdsale;  
quit;
```

This statement is similar to:  
proc print data = sashelp.prdsale;  
var country, actual run;

Country	Actual Sales
CANADA	\$925.00
CANADA	\$999.00
CANADA	\$608.00
CANADA	\$642.00
CANADA	\$656.00
CANADA	\$948.00
CANADA	\$612.00
CANADA	\$114.00
CANADA	\$685.00
CANADA	\$657.00
CANADA	\$608.00
CANADA	\$353.00
CANADA	\$107.00
CANADA	\$354.00

# Beginning Queries (cont.)

## Creating new columns:

```
proc sql;  
select predict, actual,  
predict - actual as variance  
from sashelp.prdsale;  
quit;
```

separate the formula and new column name with keyword AS

Predicted Sales	Actual Sales	variance
\$850.00	\$925.00	-75
\$297.00	\$999.00	-702
\$846.00	\$608.00	238
\$533.00	\$642.00	-109
\$646.00	\$656.00	-10
\$486.00	\$948.00	-462
\$717.00	\$612.00	105
\$564.00	\$114.00	450
\$230.00	\$685.00	-455
\$494.00	\$657.00	-163

# Beginning Queries (cont.)

## Formats and Labels:

```
proc sql;  
select predict, actual,  
       predict - actual as variance  
       format=dollar10.2 label="Variance"  
from sashelp.prdsale;  
quit;
```

specify label after column name

### Note:

the comma comes after  
label and format definitions

Predicted Sales	Actual Sales	Variance
\$850.00	\$925.00	\$-75.00
\$297.00	\$999.00	\$-702.00
\$846.00	\$608.00	\$238.00
\$533.00	\$642.00	\$-109.00
\$646.00	\$656.00	\$-10.00
\$486.00	\$948.00	\$-462.00
\$717.00	\$612.00	\$105.00
\$564.00	\$114.00	\$450.00
\$230.00	\$685.00	\$-455.00
\$494.00	\$657.00	\$-163.00
\$903.00	\$608.00	\$295.00
\$266.00	\$353.00	\$-87.00
\$190.00	\$107.00	\$83.00

# Summary Function: SUM

## Down Column:

```
proc sql;  
select sum(actual) as total  
from sashelp.prdsale;  
quit;
```

actual column is summed

```
total  
-----  
730337
```

## Across Columns:

```
proc sql;  
select sum(actual,predict) as ap  
       label="Actual + Predict"  
from sashelp.prdsale;  
quit;
```

equivalent to: actual + predict

```
Actual +  
Predict  
-----  
1775  
1296  
1454  
1175  
1302  
1434  
1329  
678  
915  
1151
```

# Summary Function: Count

## Total Number of Rows:

```
proc sql;  
select count(*) as trc  
       label="Total Row Count"  
from sashelp.prdsale;  
quit;
```

Total  
Row  
Count

-----  
1440

## Total Non-Missing Number:

```
proc sql;  
select count(region) as trc  
       label="Total Non-Missing Region Count"  
from sashelp.prdsale;  
quit;
```

Total  
Non-Missing  
Region  
Count

-----  
1440

## Total Distinct Number:

```
proc sql;  
select count(distinct region) as trc  
       label="Total Distinct Region Count"  
from sashelp.prdsale;  
quit;
```

Total  
Distinct  
Region  
Count

-----  
2

# Summary Functions (list)

SAS Function	Description
AVG,MEAN	mean of values
COUNT,FREQ,N	number of nonmissing values
CSS	corrected sum of squares
CV	coefficient of variation
MAX	largest value
MIN	smallest value
NMISS	number of missing values
PRT	probability of a greater absolute value of student's t
RANGE	range of values
STD	standard deviation
STDERR	standard error of the mean
SUM	sum of values
T	student's t value for testing the hypothesis that the population mean is zero
USS	uncorrected sum of squares
VAR	variance

# Group By Clause

## Sales by category:

```
proc sql;
select product,
       sum(actual) as tot_act
       label="Total Actual" format=dollar10.2,
       sum(predict) as tot_pred
       label="Total Predicted" format=dollar10.2
from sashelp.prdsale
group by product;
quit;
```

actual column is summed in groups

Product	Total Actual	Total Predicted
BED	\$142037.00	\$137867.00
CHAIR	\$148280.00	\$136110.00
DESK	\$149232.00	\$146195.00
SOFA	\$148588.00	\$140451.00
TABLE	\$142200.00	\$145672.00

### Reminder:

If a 'group by' clause IS NOT INCLUDED, summary functions are computed across the entire table.

If a 'group by' clause IS INCLUDED, summary functions are computed in groups

Each non-summary column selection needs to be included in the 'group by' clause.

# Having / Where Clause

```
proc sql;
select product,
       sum(actual) as tot_act
       label="Total Actual" format=dollar10.2,
       sum(predict) as tot_pred
       label="Total Predicted" format=dollar10.2
from sashelp.prdsale
group by product
having (5000 < tot_act-tot_pred)
       or (-5000 > tot_act-tot_pred);
quit;
```

Return only those products that has missed prediction by +/- \$5k.

Product	Total Actual	Total Predicted
CHAIR	\$148280.00	\$136110.00
SOFA	\$148588.00	\$140451.00

## Reminder:

If a 'group by' clause IS NOT INCLUDED, having is identical to where.

Where clause applies to each row in the table.

# Case Logic

```
CASE <column-name>  
  WHEN when-condition THEN result-expression  
  <WHEN when-condition THEN result-expression> ...  
  <ELSE result-expression>  
END AS <new column-name>
```

```
proc sql;  
select product,  
       case  
         when country in("GERMANY") then "Europe"  
         when country in("CANADA","U.S.A.") then "North America"  
         else "Unknown"  
       end as continent label="Continent",  
sum(actual) as tot_pred  
  label="Total Actual" format=dollar10.2  
from sashelp.prdsale  
group by product, continent;  
quit;
```

Product	Continent	Total Actual
BED	Europe	\$46,134.00
BED	North America	\$95,903.00
CHAIR	Europe	\$47,105.00
CHAIR	North America	\$101,175.00
DESK	Europe	\$48,502.00
DESK	North America	\$100,730.00
SOFA	Europe	\$55,060.00
SOFA	North America	\$93,528.00
TABLE	Europe	\$49,197.00
TABLE	North America	\$93,003.00

# Table Joins

## Cartesian Product:

```
proc sql;
select * from one,two;
quit;
```

## Inner Joins:

```
proc sql;
select * from one a,two b
where a.id=b.id;
quit;
```

## Outer Joins:

```
proc sql;
select * from one a
left join two b
on a.id=b.id;
quit;
```

Table One	
id	num
A	1
A	2
A	3
B	1
B	3
C	5

Table Two	
id	val
A	10
A	.
D	50
F	10

id	num	id	val
A	1	A	10
A	2	A	10
A	3	A	10
B	1	A	10
B	3	A	10
C	5	A	10
A	1	A	10
A	2	A	10
A	3	A	10
A	1	A	.
A	3	A	.
A	1	A	.
A	3	A	.
A	1	D	50
A	3	D	50
A	1	D	50
A	3	D	50
A	1	D	50
A	3	D	50
A	1	F	10
A	2	F	10
A	3	F	10
B	1	F	10
B	3	F	10
C	5	F	10

# Creating Tables

## Create a table from a query:

```
proc sql;  
create table newtable as  
  <select statement>;  
quit;
```

## Create a blank table like another:

```
proc sql;  
  create table newtable  
    like sashelp.prdsale;  
quit;
```

## Create a blank table (and insert rows into it):

```
proc sql;  
  create table avghigh  
  (  
    month character(20) label="Month",  
    temperature numeric format=7.2  
  );  
  
insert into avghigh  
  set month="January", temperature =21  
  set month="July", temperature=84  
;  
quit;
```

# Some Useful Queries:

Count non-missing values in SQL quickly:

```
proc sql;  
  select  
    count(*) as total,  
    count(id) as id_ct,  
    count(val) as val_ct  
  from two;  
quit;
```

Counts total number of rows

Counts total number of  
non-missing values  
(column wise)

total	id_ct	val_ct
4	4	3

# Coalesce

```
proc sql;
  select a.key, a.id,
         coalesce(a.num, b.val, c.nb, 0) as special_number
  from one a
  left join two b
        on a.key=b.key
  left join three c
        on a.key=c.key;
quit;
```

Picks first non-missing value

Table One	
key id	num
1 B	3
2 C	.

Table Two	
key id	val
1 A	10
2 A	.

Table Three	
key id	nb
1 A	10
2 A	500
3 D	.
4 D	50
5 F	10

key id	special_
	number
1 B	3
2 C	500



# Useful PROC SQL Options

- feedback
  - log displays standardized SQL
- stimer
  - Amount of time used in each statement
- inobs / outobs
  - Limit the input/output to the first n observations
- distinct
  - Only selects unique values
- noexec / validate
  - don't execute the code
    - noexec applies to entire proc sql procedure
    - validate applies to only next query

# Macro Variables

```
proc sql noprint;  
select country, sum(actual) into :c1-:c3, :s1-:s3  
from sashelp.prdsale  
group by country;  
quit;
```

Country	
-----	
CANADA	246990
GERMANY	245998
U.S.A.	237349

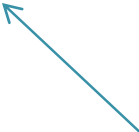
Normal query output (with  
noprint removed)



```
%put &c1 had total sales of %sysfunc(putn(&s1,dollar8.)).;
```

CANADA had total sales of \$246,990.

Printed in the SAS log.



# Compare data to an average...

```

proc sql;
%let whereclause =
  where product="SOFA" and country="GERMANY"
  and region="WEST" and division="CONSUMER";

select avg(actual) into :ma from sashelp.prdsale
&whereclause;

title "Germany consumer sofa sales dates in the west region where";
title2 "actual sales were greater than the average of "%sysfunc(putn(&ma,dollar4.))".";
select year, month,
  actual
    format=comma4.0
    label="Actual $ Sales",
  actual-predict as amp
    label="Actual - Predicted $ Difference"
    format=negparen4.0,
  actual / &ma -1 as per
    label="% above average"
    format=percent8.2
from sashelp.prdsale
&whereclause
and actual > &ma
order by year desc, month desc;
title;title2;
quit;

```

Germany consumer sofa sales dates in the west region where  
actual sales were greater than the average of \$633.

Year	Month	Actual - Predicted		% above average
		\$ Sales	\$ Difference	
1994	Nov	996	353	57.38%
1994	Sep	763	452	20.56%
1994	May	866	458	36.84%
1994	Apr	768	287	21.35%
1994	Mar	847	709	33.83%
1994	Feb	661	25	4.44%
1993	Dec	906	(43)	43.16%
1993	Aug	913	562	44.26%
1993	Jul	705	520	11.40%
1993	Jun	900	871	42.21%
1993	May	860	477	35.89%
1993	Apr	876	605	38.42%
1993	Mar	749	(20)	18.35%
1993	Feb	748	-106	18.19%
1993	Jan	872	(51)	37.78%

# Struggling with a SQL question?





# Thanks!

- Minne JMP User Group Meeting
  - January 21<sup>st</sup>
  - Check out <http://tcasug.org/jmp> for more information!
  - Contact Mike
    - olson [dot] mike [dot] p [at no spam] gmail [dot] com

# Code backup:

```
proc sql;  
select * from sashelp.prdsale;  
quit;
```

```
proc sql;  
select country,actual  
from sashelp.prdsale;  
quit;
```

```
proc sql;  
select predict, actual,  
predict - actual as variance  
from sashelp.prdsale;  
quit;
```

```
proc sql;  
select predict, actual,  
predict - actual as variance  
format=dollar10.2 label="Variance"  
from sashelp.prdsale;  
quit;
```

```
proc sql;  
select sum(actual,predict) as ap  
label="Actual + Predict"  
from sashelp.prdsale;  
quit;
```

```
proc sql;  
select count(*) as trc  
label="Total Row Count"  
from sashelp.prdsale;  
quit;
```

```
proc sql;  
select count(region) as trc  
label="Total Non-Missing Region Count"  
from sashelp.prdsale;  
quit;
```

```
proc sql;  
select count(distinct region) as trc  
label="Total Distinct Region Count"  
from sashelp.prdsale;  
quit;
```