

22S:166
SAS

Queries and SQL
More on Data Integrity

Lecture 20
Nov. 26, 2007

Kate Cowles
374 SH, 335-0727
kcowles@stat.uiowa.edu

Structured Query Language

- query: a view of data which represents the data from one or more tables
- queries built in a relational database using Structured Query Language or SQL
- SQL is the standard language for relational databases
- includes the capability of manipulating both the structure of a database and its data
- most common use: to create a simple SELECT query

Proc sql in SAS

- SAS data files and SQL tables
 - structure of an SQL table is very similar to that of a SAS data file
 - only difference: SASdata file has inherent ordering
 - in SAS System, SQL table is represented as a SASdata file
- *proc sql* can perform some of the operations provided by the *data* step and the *print*, *sort*, and *means* procedures
 - often can achieve same results as these procedures with fewer and short statements
 - why should you still know how to do these tasks with *print*, *sort*, *means*, etc.?
 - * because you are likely to have to maintain or modify older programs written before *proc sql* was added to SAS

Queries using proc sql select statement

- *select* statement in *proc sql* finds and displays specified records and variables
- can also link files, calculate summary statistics, sort, etc.

Return to sites and deposition example

```
options linesize = 75 pagesize = 60 nodate nonumber ;

data depo ;
infile 'depoRep90s.asp' firstobs = 8 ;
input SiteID $ Per $8. Year Crit1 Crit2 Crit3 Crit4 Ca Mg
K Na NH4 NO3 InorgN Cl SO4 HLab HField Svol Ppt Pct ValidF ValidL
Days @196 Date1 mmddyy10. @209 Date2 mmddyy10. ;
drop Per Crit1-Crit4 Ca Mg K Na NH4 NO3 InorgN Cl HLab HField
Svol Ppt Pct ValidF ValidL ;
daysop = Date2 - Date1 ;
format Date2 Date1 date8. ;
run ;

data sites ;
infile '/space/kcowles/166/lectures/lect1mkc/stateC0.asp' firstobs = 19
missover ;
input @13 SiteID $ @20 sitename $18. @40 strtdate mmddyy10. @53 stopdate ;
if strtdate ne . ; * subsetting if: exclude observations meeting condit:
format strtdate stopdate date8. ;
drop sitename ;
run ;
```

```
proc sql ;
title 'Proc sql listings' ;
select * from sites ; /* list all variables and records *,
```

```
Proc sql listings
SiteID  strtdate  stopdate  elev
-----
C000    22APR80     .         2298
C001    04OCT83     .         1213
C002    05JUN84     .         3520
C008    29DEC87     .         2502
C010    02FEB99     .         2926
C015    20MAR79     .         1998
C019    29MAY80     .         2490
C021    17OCT78     .         2362
C022    22MAY79     .         1641
C091    26MAY92     .         3292
C092    13JAN88     .         3206
C093    14OCT86     .         2527
C094    04NOV86     .         2524
C095    29JUL86     02JAN90   2758
C096    29JUL86     .         3249
C097    07FEB84     .         3234
C098    16AUG83     .         3159
C099    28APR81     .         2172
```

- sites file
 - SiteID
 - strtdate
 - stopdate
 - elev
 - SiteID
- depo file
 - SiteID
 - Year
 - SO4

```
select SiteID, elev from sites ; /* list selected variables, all rec:
```

```
Proc sql listings
SiteID  elev
-----
C000    2298
C001    1213
C002    3520
C008    2502
C010    2926
C015    1998
C019    2490
C021    2362
C022    1641
C091    3292
C092    3206
C093    2527
C094    2524
C095    2758
C096    3249
C097    3234
C098    3159
C099    2172
```

```
* multiple-table query ;

title2 'Multiple table query' ;
select s.siteID, s.elev, d.S04, d.Year
from sites s, depo d
where s.SiteID = d.SiteID
order by s.SiteID ;
```

Multiple table query

SiteID	elev	S04	Year
C000	2298	1.2	1992
C000	2298	1.28	1998
C000	2298	1.07	1996
C000	2298	2.08	1991
C000	2298	1.01	1999
C000	2298	1.18	2000
C000	2298	1.1	1997
C000	2298	1.5	1993
C000	2298	1.46	1995
C000	2298	1.31	1994
C001	1213	3.64	1995
C001	1213	3.09	1992
C001	1213	2.98	1994
C001	1213	2.3	1993
C001	1213	2.44	1998
C001	1213	2.53	1997
C001	1213	3.19	1991
C001	1213	2.06	2000
.	.	.	.
.	.	.	.
.	.	.	.

```
* more sophisticated query ;

title2 'More complicated SELECT and ORDER' ;
select s.siteID, s.elev, d.S04, d.Year
from sites s, depo d
where s.SiteID = d.SiteID and d.Year > 1995
order by s.SiteID, d.Year ;
```

More complicated SELECT and ORDER

SiteID	elev	S04	Year
C000	2298	1.07	1996
C000	2298	1.1	1997
C000	2298	1.28	1998
C000	2298	1.01	1999
C000	2298	1.18	2000
C001	1213	2.99	1996
C001	1213	2.53	1997
C001	1213	2.44	1998
C001	1213	3.49	1999
C001	1213	2.06	2000
C002	3520	14.82	1996
C002	3520	10.9	1997
C002	3520	8.67	1998
C002	3520	10.7	1999
C002	3520	19.32	2000
.	.	.	.
.	.	.	.
.	.	.	.

```
* summing and grouping ;

title2 'Total Deposition' ;

select siteID, sum(S04) as totso4
from depo
group by SiteID
order by SiteID
;
```

Proc sql listings
Total Deposition

SiteID	totso4
C000	13.19
C001	28.71
C002	115.68
C008	26.25
C010	4.85
C015	23.28
C019	26.17
C021	35.49
C022	30.16
C091	69.82
C092	33.01
C093	65.73
C094	40.81
C096	39.53
C097	82.16
C098	57.27
C099	39.88

Producing report with proc means

```
proc means data = depo ;
class SiteID ; /* separate summary stats by this variable */
var S04 ; /* which numeric variable to summarize */
output out = meandepo mean=avgso4 ; /* identify output dataset and
variable name for summary stat */
run ;

proc print data = meandepo ;
run ;
```

Obs	Site ID	_TYPE_	_FREQ_	avgso4
1		0	161	4.5465
2	C000	1	10	1.3190
3	C001	1	10	2.8710
4	C002	1	10	11.5680
5	C008	1	10	2.6250
6	C010	1	2	2.4250
7	C015	1	10	2.3280
8	C019	1	10	2.6170
9	C021	1	10	3.5490
10	C022	1	10	3.0160
11	C091	1	9	7.7578
12	C092	1	10	3.3010
13	C093	1	10	6.5730
14	C094	1	10	4.0810
15	C096	1	10	3.9530
16	C097	1	10	8.2160
17	C098	1	10	5.7270
18	C099	1	10	3.9880

```

title 'Report produced using proc means' ;

proc print data = meandepo noobs ;
var SiteID avgso4 ;
where _type_ = 1 ;
run ;

```

Report produced using proc means

Site ID	avgso4
C000	1.3190
C001	2.8710
C002	11.5680
C008	2.6250
C010	2.4250
C015	2.3280
C019	2.6170
C021	3.5490
C022	3.0160
C091	7.7578
C092	3.3010
C093	6.5730
C094	4.0810
C096	3.9530
C097	8.2160
C098	5.7270
C099	3.9880

Types of data integrity

- “data integrity” is database language for data validity and checking
- types
 - entity integrity: no duplicate rows
 - domain integrity: values in any given column fall within an acceptable range or set
 - referential integrity: foreign key values point to valid rows in the referenced table
 - user-defined integrity: data complies with other rules specific to the application