Chapter 19. Introduction to PROC SQL

19.1. Standard Query Language (SQL)

- Standard language for relational database management systems
- Communicate with a database
  - Update data on a database.
  - Retrieve data from a database.
  - Request information from database to answer questions
- Common database management systems: Oracle, Sybase, Microsoft SQL Server, Access
- Standard commands: Select, Insert, Update, Delete, Create, Drop

19.2. PROC SQL

- Base SAS procedure: Combine the functionality of DATA and PROC steps in a *single* step.
  - Sort, summarize, subset, merge, and concatenate datasets.
  - Create new variables, or produce a new table.
  - Retrieve, update and report on information from SAS datasets or other database.
- PROC SQL can do the same task with fewer and shorter statements than traditional SAS code.
- It often uses fewer resources than conventional DATA and PROC steps.
- SAS has fewer data types than standard SQL.
  - Character
  - Numeric (numeric, decimal, integer, smallint, float, real, double, precision, and date)
- PROC SQL follows the guidelines set by the American National Standards Institute (ANSI).
- A SQL view is a stored SELECT statement executed at run time. (cf) NOPRINT)

General 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;
```

- SQL statement
  - You can have as many SQL statements as you want in a single PROC SQL.

SQL Statement	Description
ALTER TABLE	Add, drop, and modify columns in a table.
CREATE	Build new tables, views, or indexes.
DELETE	Eliminate unwanted rows from a table or view.
DESCRIBE	Display table and view attributes.
DROP	Eliminate entire tables, views, or indexes.
INSERT	Add rows of data to tables or views.
RESET *<options>*	Add to or change PROC SQL options without re-invoking the procedure.
UPDATE	Modify data values in existing rows of a table or view.
JOIN *tables* on *variable(s)*	Merge datasets based on certain variable(s).

Example

Raw Data

sports

Obs	CustomerID	Name	Address
1	101	Murphy's Sports	115 Main St.
2	102	Sun N Ski	2016 Newberry Ave.
3	103	Sports Outfitters	19 Cary Way
4	104	Cramer & Johnson	4106 Arlington Blvd.
5	105	Sports Savers	2708 Broadway

pbc1

Obs	ID	Treatment	Age	Gender	Stage
1	1	1	58.7652	1	4
2	2	1	56.4463	1	3
3	3	1	70.0726	0	4
4	4	1	54.7406	1	4
5	9	1	42.5079	1	2

pbc2

Obs	ID	Ascites	Hepato	Spiders	Bili	Chol	Albu	Copp	Alka	SGOT	Trig	Platelet	Protime
1	1	1	1	1	14.5	261	2.6	156	1718	137.95	172	190	12.2
2	2	0	1	1	1.1	302	4.14	54	7394.8	113.52	88	221	10.6
3	3	0	0	0	1.4	176	3.48	210	516	96.1	55	151	12
4	4	0	1	1	1.8	244	2.54	64	6121.8	60.63	92	183	10.3
5	5	0	1	1	3.4	279	3.53	143	671	113.15	72	136	10.9

SAS Code

```
/* Create a table + Print */
proc sql;
create table work.sports0
(CustomerID num, Name char(17), Address char(20));
insert into work.sports0
 values (101, "Murphy's Sports", "115 Main St.")
 values (102, "Sun N Ski", "2016 Newberry Ave.")
 values (103, "Sports Outfitters", "19 Cary Way")
 values (104, "Cramer & Johnson", "4106 Arlington
Blvd.");
select * from work.sports0; quit;
```

Output

CustomerID	Name	Address
101	Murphy's Sports	115 Main St.
102	Sun N Ski	2016 Newberry Ave.
103	Sports Outfitters	19 Cary Way
104	Cramer & Johnson	4106 Arlington Blvd.

```
/* Concatenate two tables */
* Another table;
proc sql;
create table sports00
(CustomerID num, Name char(13), Address char(13));
insert into sports00
 values (105, "Sports Savers", "2708 Broadway");
quit;
* Concatenate + print;
proc sql;
create table sports as
select * from sports0
union all
select * from sports00;
select name, address from sports; quit;
```

Name	Address
Murphy's Sports	115 Main St.
Sun N Ski	2016 Newberry Ave.
Sports Outfitters	19 Cary Way
Cramer & Johnson	4106 Arlington Blvd.
Sports Savers	2708 Broadway

```
/* Read an existing table */
proc sql;
select name, address from sports
where customerID = 102; quit;
```

Name	Address
Sun N Ski	2016 Newberry Ave.

```
/* Create a new table (+ sort) + print */
proc sql;
create table sports3 as
select *, customerID - 100 as SimpleID,
substr(Name,1,1) as Initial
from sports
order by name;
select SimpleID, Initial, Address
from sports3; quit;
```

SimpleID	Initial	Address
4	C	4106 Arlington Blvd.
1	M	115 Main St.
3	S	19 Cary Way
5	S	2708 Broadway
2	S	2016 Newberry Ave.

```sql
/* Merge two tables */
proc sql;
create table pbc as
select *
from pbc1, pbc2
where pbc1.id = pbc2.id
order by pbc1.id;
select id, age, stage, hepato, albu
from pbc where id <= 5; quit;
```

ID	Age	Stage	Hepato	Albu
1	58.7652	4	1	2.6
2	56.4463	3	1	4.14
3	70.0726	4	0	3.48
4	54.7406	4	1	2.54
5	38.1054	3	1	3.53

```sql
/* Rename, Label, Format, New variable */
proc format;
value mffmt 0 = "Male"
            1 = "Female";
run;

proc sql;
create table pbc1_new as
select ID label = "Patient ID",
       stage as pbcstage,
       age format = 5.1,
       round(age) as age2,
       gender format = mffmt.
from pbc1;
select *
from pbc1_new
where pbcstage = 1; quit;
```

Patient ID	Stage	Age	age2	Gender
52	1	50.5	51	Male
58	1	44.6	45	Male
65	1	40.2	40	Female
98	1	28.9	29	Female
102	1	56.6	57	Female
153	1	49.6	50	Female
174	1	55.6	56	Female
206	1	62.0	62	Female
218	1	34.6	35	Female
258	1	51.5	51	Female
272	1	38.4	38	Female
285	1	46.3	46	Female
61	1	43.9	44	Male
73	1	38.5	38	Female
107	1	62.5	63	Female
150	1	35.0	35	Female

```
/* Having, Group by */
proc sql;
select ID, age, stage, hepato
from pbc
where Trig <= 200 AND 3.5 <= Albu <= 6
group by stage, hepato, id
having 11 <= Protime <= 14;
quit;
```

ID	Age	Stage	Hepato
61	43.8987	1	0
73	38.4942	1	0
153	49.6044	1	0
206	61.9904	1	0
25	45.0732	2	0
90	33.4757	2	0
93	36.5339	2	0
104	43.0171	2	0
135	42.9678	2	0
89	52.4435	2	1