Chapter 5

Creating and Managing Views Using PROC SQL

What is a View?

- A view is a stored query that is executed when the view is used.
- The view does not contain the data, just the logic for accessing the data.
- Views can be used in a SAS procedure, data step, or function.
- Views can be joined with tables or other views.

Why Are Views Useful

- Often save space
- Ensure input data sets are always current, because data is derived from tables at execution time
- Hide confidential columns (e.g., SSN) while allowing access to view other columns in the same table

Computer Exercise 1

SSN	Major Code	Degree	Grade
123-00-5526	135	Bachelors	9
206-00-1454	135	Bachelors	10
323-00-2746	126	Masters	9
367-00-9580	126	Bachelors	9
564-00-6508	126	Bachelors	9
782-00-8284	135	Masters	10
354-00-9603	126	Masters	10
590-00-6170	135	Bachelors	8
060-00-2046	126	Bachelors	9
983-00-6455	135	Masters	9
638-00-0350	135	Bachelors	9
861-00-4706	126	Bachelors	9

Homework 1

SSN	Major Code	Degree	HW 1
367-00-9580	126	Bachelors	19
564-00-6508	126	Bachelors	19
060-00-2047	126	Bachelors	17
123-00-5526	135	Bachelors	20
206-00-1454	135	Bachelors	19
638-00-0350	135	Bachelors	19
861-00-4706	126	Bachelors	20
983-00-6455	135	Masters	19
782-00-8284	135	Masters	19
354-00-9603	126	Masters	19
590-00-6170	135	Bachelors	18

Creating a View

```
proc sql;
 create view viewname  as
 select column1, column2,…, columnn
 from table1
 <where expression>
<group by column1,…,columnn>
<having expression>
<order by column1,…,columnn>;
```

*The where, group by, having, and order by clauses are optional.

Creating a View

Inner view:

```
select ce1.code label="Major
    Code",ce1.degree
    label="Degree",ce1.grade as ce1
    label="Computer Exercise 1",
    hw1.grade as hw1 label="Homework
    1" from ce1 left join hw1 on
    ce1.ssn=hw1.ssn
```

Creating a View

Outer clause:

```
proc sql;
select code,avg(ce1) as ceavg
    label="CE 1" format=4.1, avg(hw1)
    as hw1avg label="HW 1" format=5.1
    from . . .
group by code;
quit;
```

Creating a View

Create and execute the view:

```
proc sql;
create view Major_Comp as . . .;
quit;

proc sql;
select * Major_Comp;
quit;
```

Using a View

Views can also be used in PROC steps:

```
proc sql;
create view JoinGrades as select
    ce1.code label="Major
    Code",ce1.degree
    label="Degree",ce1.grade as ce1
    label="Computer Exercise 1",
    hw1.grade as hw1 label="Homework
    1" from ce1 left join hw1 on
    ce1.ssn=hw1.ssn
    order by code;quit;
```

Using a View

Views can also be used in PROC steps:

```
proc means data=JoinGrades
   maxdec=2; by code;
var ce1 hw1;
run;
```

Describe View Statement

- Use a DESCRIBE VIEW statement to display the definition of a view in the SAS log.

```
proc sql;
   describe view Major_Comp
    JoinGrades;
   quit;
```

Guidelines for Using Views

- Avoid using the ORDER BY clause in a view definition, otherwise the data will have to be sorted each time the view is executed.
- It is more efficient to create a table if the same data is used many times in one program

Guidelines for Using Views

- Avoid creating views that are based on tables whose structure (e.g., columns in table) may change
- Specify a one level name (e.g., claims, not work.claims) in the FROM clause if a view resides in the same SAS library as the contributing table(s)

USING LIBNAME

- References to a single-level table name assume the table is in the same library as the view
- USING LIBNAME can be appended to the CREATE VIEW clause to resolve confusion in table references

USING LIBNAME

```
proc sql;
create view viewname as
select * from libname.table1
using libname 'directory';
```

Updating a View

- Underlying tables can be updated with UPDATE, INSERT, and DELETE
- Limitations
  - Only a single table can be updated
  - Views with WHERE clauses can be updated
  - Views with ORDER BY, HAVING or GROUP BY cannot be updated

Dropping a View

- To drop (delete) a view, use the drop view statement.

```
proc sql;
   drop view viewname;
   quit;
```