Managing PROC SQL Views

- A view is…(recall the definition and property covered before)

- An example:

```
proc sql;
   create view sasuser.faview as
      select lastname, firstname, gender,
             int((today()-dateofbirth)/365.25) as Age,
             substr(jobcode,3,1) as Level,
             salary
        from sasuser.payrollmaster,
             sasuser.staffmaster
        where jobcode contains 'FA' and
              staffmaster.empid=
              payrollmaster.empid;
```

A view can be used in other SAS procedure and DATA steps

```
proc tabulate data=sasuser.faview;
    class level;
    var age;
    table level*age*mean;
run;
```

Level		
1	2	3
Age	Age	Age
Mean	Mean	Mean
50.00	54.06	55.57

One-level name in the FROM clause

```
proc sql;
    create view sasuser.payrollv as
    select *
    from payrollmaster;
quit;
```

```
proc sql;
    create view sasuser.payrollv as
    select *
    from sasuser.payrollmaster;
quit;
```

- The default libref for the table or tables in the FROM clause is the libref of the library that contains the view.
- Using a one-level name in the FROM clause prevents you from having to change the view if you assign a different libref to the SAS data library that contains the view and its contributing table or tables.

Using an Embedded LIBNAME Statement

- As an alternative to omitting the libref in the FROM clause, you can embed a LIBNAME statement in a USING clause to store a SAS libref in a view. Embedding a LIBNAME statement is a more flexible approach because
  - it can be used regardless of whether the view and the underlying tables reside in the same library;
  - it avoids the confusion that might arise if a libref is omitted from a table name in the FROM clause.
- An embedded LIBNAME statement can only be used with a PROC SQL view.
- The USING clause must be the **last** clause in the CREATE VIEW statement (before the Quit statement).

An Example

```
libname airline 'SAS-library-one';
…
proc sql;
    create view sasuser.payrollv as
    select*
    from airline.payrollmaster
    using libname airline 'SAS-library-two';
quit;

proc print data=sasuser.payrollv;
run;
```

In this example, while the view *Sasuser.Payrollv* is executing in the PROC PRINT step, the libref *Airline is dynamically assigned* in the USING clause. This overrides the earlier assignment of the libref in the LIBNAME statement for the duration of the view's execution. After the view executes, the original libref assignment is re-established and the embedded assignment is cleared.

Updating views

- You can update the data underlying a PROC SQL view using the INSERT, DELETE, and UPDATE statements under the following conditions:
  - You can only update a single table through a view. The table cannot be joined or linked to another table, nor can it contain a subquery.
  - You can update a column using the column's alias, but you cannot update a derived column (a column that is produced by an expression).
  - You can update a view that contains a WHERE clause. The WHERE clause can be specified in the UPDATE clause or in the view. You cannot update a view that contains any other clause such as an ORDER BY or a HAVING clause.
  - You cannot update a summary view (a view that contains a GROUP BY clause).
- Updating a view does *not* change the stored instructions for the view. Only the data in the underlying table is updated.

An Example

```
proc sql;
   create view sasuser.raisev as
      select empid, jobcode,
             salary format=dollar12.,
             salary/12 as MonthlySalary
             format=dollar12.
        from payrollmaster;

proc sql;
   select *
      from sasuser.raisev
      where jobcode in ('PT2','PT3');
```

EmpID	JobCode	Salary	MonthlySalary
1333	PT2	$124,048	$10,337
1404	PT2	$127,926	$10,661
1118	PT3	$155,931	$12,994
1410	PT2	$118,559	$9,880
1777	PT3	$153,482	$12,790
1106	PT2	$125,485	$10,457
1442	PT2	$118,350	$9,863
1478	PT2	$117,884	$9,824
1890	PT2	$120,254	$10,021
1107	PT2	$125,968	$10,497
1830	PT2	$118,259	$9,855
1928	PT2	$125,801	$10,483

```
proc sql;
   update sasuser.raisev
      set salary=salary * 1.20
      where jobcode='PT3';
```

```
116   proc sql;
117      update sasuser.raisev
118         set salary=salary * 1.20
119         where jobcode='PT3';
NOTE: 2 rows were updated in SASUSER.RAISEV.
```

Result

```
proc sql;
   select *
      from sasuser.raisev
      where jobcode in ('PT2','PT3');
```

EmpID	JobCode	Salary	MonthlySalary
1333	PT2	$124,048	$10,337
1404	PT2	$127,926	$10,661
1118	PT3	$187,117	$15,593
1410	PT2	$118,559	$9,880
1777	PT3	$184,178	$15,348
1106	PT2	$125,485	$10,457
1442	PT2	$118,350	$9,863
1478	PT2	$117,884	$9,824
1890	PT2	$120,254	$10,021
1107	PT2	$125,968	$10,497
1830	PT2	$118,259	$9,855
1928	PT2	$125,801	$10,483

Practice on PROC SQL Views

- Business scenario: Tom is a sales manager (Manager_ID=120102) who frequently needs access to personnel information for his direct reports, including name, job title, salary, and years of service.
- The data Tom needs can be obtained from these tables: sasuser.Employee_Addresses, sasuser.Employee_Payroll, and sasuser.Employee_Organization.
- Create a view, Tom_V, containing personnel information for Tom's direct reports to provide the information that Tom needs while avoiding unintended access to data for employees who do not report to him. In your view, use an alias "Name" and format=$25. for Employee_Name, and an alias "Title" and format=$15. for Job_Title; use a label "Annual Salary" and format=comma10.2 for Salary; define a column called "YOS" to hold the values of the years of service you calculate, and label it as "Years of Service".
- Display the contents of the view you just created, and sort the values by Title and then by YOS both in descending order.

The code and result

Name	Title	Annual Salary	Years of Service
Nowd, Fadi	Sales Rep. IV	30,660.00	32
Phoumirath, Lynelle	Sales Rep. IV	30,765.00	30
Hofmeister, Fong	Sales Rep. IV	32,040.00	25
Kletschkus, Monica	Sales Rep. IV	30,890.00	10
Platts, Alexei	Sales Rep. IV	32,490.00	6
Comber, Edwin	Sales Rep. III	28,345.00	42
Hayawardhana, Caterina	Sales Rep. III	30,490.00	42
Kaiser, Fancine	Sales Rep. III	28,525.00	38
Kierce, Franca	Sales Rep. III	28,745.00	18
Pilgrim, Daniel	Sales Rep. III	36,605.00	18
Roebuck, Alvin	Sales Rep. III	30,070.00	17
Kingston, Alban	Sales Rep. III	28,830.00	14
Tannous, Cos	Sales Rep. III	28,135.00	10
Iyengar, Marina	Sales Rep. III	29,715.00	10
...	...	...	...

Practice on PROC SQL Views, Con't

- Use Tom_V to produce simple descriptive statistics: minimum, mean and maximum salaries of different job titles. You need to use two methods to achieve the same result , exactly as shown in the following table.
  - SAS PROC MEANS procedure (the PROC SQL view is used as a SAS dataset)
  - PROC SQL procedure

Analysis Variable : Salary Annual Salary				
Title	N Obs	Minimum	Mean	Maximum
Sales Rep. I	18	25185.00	26466.67	27260.00
Sales Rep. II	13	26165.00	27123.46	28100.00
Sales Rep. III	12	28135.00	29895.42	36605.00
Sales Rep. IV	5	30660.00	31369.00	32490.00