Class Exercise 3

This exercise is based upon Chapter 3 of the SAS Certification Prep Guide. We will experiment with in-line views in the following exercise. If the output data sets are too large to display in your assignment, use the OUTOBS= option in PROC SQL to limit their size (OUTOBS=15 would be fine). It may also be useful to place each of the SELECT steps in an in-line view so that the COUNT(\*) statement can be used to obtain the size of each output data set.

Let’s work once more with the Admissions data. Import all three Excel spreadsheets into SAS (name them **Fall2008**, **Fall2009**, and **Fall2010**).

I. Multiple nested queries

Suppose we wanted to identify all first-year students from 2008 who lived on campus their first year, and were listed as CAS majors in their second year (2009). We could identify first-year on-campus students with the following code:

**proc** **sql**;

title "2008 cohort subsample on campus";

select noobs label="ID", school from work.fall2008

 where class='Freshman'

 and housing=**1**;

 **quit**;

And we could identify 2009 CAS majors with the following code:

**proc** **sql**;

title "2008 cohort subsample CAS majors in 2009";

select noobs label="ID" from work.fall2009

 where school='College of Arts and Sciences';

 **quit**;

Treating the first piece of code as an in-line view, we can combine the statements to complete our view. Note that we include 2009 GPA in our view, though we will need to drop it for later queries.

**proc** **sql**;

select noobs label="ID", cltotgpa as GPA from work.fall2009 as f09

where school='College of Arts and Sciences'

and f09.noobs in

(select noobs from work.fall2008

 where class='Freshman'

 and housing=**1**);

 **quit**;

We can expand the query to identify on-campus first-year students who were CAS majors in 2009 and 2010 and had 2010 GPA’s greater than 3.5. You may need to replace “cltotgpa gt 3.5” with “input(cltotgpa,5.) gt 5” if cltotgpa was uploaded as a character variable rather than a numeric variable. This code focuses on the 2010 data:

**proc** **sql**;

title;

select noobs label="ID", cltotgpa as GPA from work.fall2010 as f10

where school='College of Arts and Sciences' and cltotgpa gt **3.5**;

**quit**;

We can include two nested in-line views to complete our inquiry:

**proc** **sql**;

select noobs label="ID", cltotgpa as GPA from work.fall2010 as f10

where school='College of Arts and Sciences' and cltotgpa gt **3.5** and f10.noobs in

(select noobs label="ID" from work.fall2009 as f09

where school='College of Arts and Sciences'

and f09.noobs in

(select noobs from work.fall2008

 where class='Freshman'

 and housing=**1**));

**quit**;

II. Joins of a table and an in-line view

Working with the same data sets, we select the ID numbers and home state of 2008 on-campus first-year students from outside South Carolina using an in-line view. Rather than use the in-line view as we have in other examples, we combine it with a table (fall2009) in an inner join. In addition to the ID numbers and home state of 2008 first-year students, we also find their 2009 GPAs.

Note the following feature of the code--we use an alias for the in-line view (f08Fresh) to simplify our **select** and **where** clauses.

**proc** **sql**;

select f08Fresh.noobs, f08Fresh.stcd, f09.cltotgpa from work.fall2009 as f09, (select noobs, stcd from work.fall2008

 where class='Freshman' and housing=**1**

 and stcd ne 'South Carolina') as f08Fresh

where f09.noobs=f08Fresh.noobs;

**quit**;

We construct a full join with the same table and in-line view using a **coalesce** statement. Grad students: comment on what happens if you select only f08Fresh.noobs without using **coalesce**.

**proc** **sql**;

select coalesce(f08Fresh.noobs, F09.noobs) as ID, coalesce(f08Fresh.stcd, f09.stcd) as stcd, f09.cltotgpa from (select noobs, stcd from work.fall2008

 where class='Freshman' and housing=**1**

 and stcd ne 'South Carolina') as f08Fresh full join work.fall2009 as f09

on f09.noobs=f08Fresh.noobs;

**quit**;