SAS Refresh!

15 Questions

Q. 01/15

How to connect to a directory (folder)?

```
libname reference-name “Path”;
```

http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a000214133.htm

Q. 02/15

How do you write SQL statement in SAS?

```
/* Wrap the SQL statement with PROC SQL */
proc sql;
    create table library-name.data-name as
    select column-1, column2
    from library-name.data-name
    where condition
    order by column-1, column2; << put ; here.
quit;
```

Q. 03/15

Explain what is inner join, left join and right join?

INNER JOIN: returns rows when there is a match in both tables.
LEFT JOIN: returns all rows from the **left** table, even if there are no matches in the **right** table.
RIGHT JOIN: returns all rows from the **right** table, even if there are no matches in the **left** table.

Q. 04/15

What is a data step?
Give a simple data step example (any).

A **DATA step** is a group of SAS statements that begin with a data statement that manipulate existing SAS data sets or create SAS data sets from raw data files.

```
/* Combine dataset a and b together into c*/
data c;
   /* Note there is no comma here, SAS is weird */
   Set a b;
run;
```

https://v8doc.sas.com/sashtml/lrcon/z1081414.htm

Q. 05/15

What is a proc step?
Give a simple proc step example (any).

A group of SAS **procedure statements** is called a PROC step. The PROC step consists of the following: a beginning procedure (PROC) statement with options. typically, statements specifying plot types, variables, and options. an ending RUN statement.

```
/* Sorting procedure */
proc sort
   data = library-name.data-name
   out = library-name.sorted-data-name;
   by column-name column-name2;
run;

http://support.sas.com/kb/24/835.html
```

Q. 06/15

How to you add a row number?

One way is to use automatic variable _n_ with data step

```
/* Adding row number with data step */
data data-name;
   set data-name;
   rownum = _n_;
run;
```

http://documentation.sas.com/?docsetId=lrcon&docsetTarget=p0e0mk25gs9binn1s9jiu4otau29.htm&docsetVersion=9.4&locale=en

Q. 07/15

How to create a macro variable?
How to print to the screen?

```
/* Define a macro variable with %let statement */
%let myvariable = hello world;

/* Print to the screen using %put statement,
   note to apply the variable, add & in front of the name */
%put &myvariable;
```

Q. 08/15

How to convert string to number?
How to convert number to string?

```
/* Convert string “1000” to number 1000 */
new_number = put(“1000”, 4.);

/* Convert number 1000 to string “1000” */
new_string = input(1000, $4.);

Tip to remember: put function has 3 character
                 num << convert to number
```

https://blogs.sas.com/content/sgf/2015/05/01/converting-variable-types-do-i-use-put-or-input/

Q. 09/15

Just to double check
What does put and input function do??

Put() function takes string and convert to number.
Input() function takes number and convert to string.

99% of SAS programmer gets this confused on a daily basis.

Put
Num << Put, Return you a (Number), Put string to Num.
Put has 3 letters, so is Num.
Think of Put() as Num() function!
Input() as String() function!

Q. 10/15

How do I format date?
PROC SQL or Data Step.

```
/* Format date column */
proc sql;
   create table library-name.data-name as
   select date-column format=mmddyy10.
   from library-name.data-name;
quit;

/* Commonly used date format */
/* mmddyy10 = 01/01/2019 */
/* date9 = 01Jan2019 */
```

https://communities.sas.com/t5/SAS-Procedures/How-to-format-date-values-in-Proc-SQL/td-p/228015

Q. 11/15

How to import / export excel spreadsheet?

```
/* Using proc step to import & export */
proc import
    datafile = “file-path”
    out = data-name
    dbms = excel
    replace;
run;

proc export
    outfile = ‘file-path’
    data = data-name
    dbms = excel
    replace;
run;
```

Q. 12/15

How to concat strings?
column_1 = A, column_2 = B, column_3 = C
I want new column as A-B-C

```
/* Using concat function */
proc sql;
   create table data-name as
   select catx(‘-’, column_1, column_2, column_3) as string_1
   from some-table;
quit;

/* or using ||, but they have to be all string type! */
proc sql;
   create table data-name as
   select column_1 || ‘-’ || column_2 || ‘-’ || column_2 as
string_2
   from some-table;
quit;
```

https://communities.sas.com/t5/SAS-Procedures/PROC-SQL-concatenate-column-with-string/td-p/300143

Q. 13/15

How to filter for date_col >= 01/01/2019?
where <insert here>

```
/* SAS is very picky about date */
proc sql;
   create table data-name as
   select column-1, column-2
   from some-table
   where date >= ‘01JAN2019’d;
   /* where date >= ‘01-01-2019’d;
      will not work, SAS does not understand.
      Must be in date9 or date7 format. */
quit;
```

Q. 14/15

How to filter for
date_col from 01/01/2019 to 12/31/2019
where <insert here>

```
/* date-column between … and … ; */
proc sql;
   create table data-name as
   select column-1, column-2
   from some-table
   where date between ‘01JAN2019’d and ‘31DEC2019’d;
quit;
```

Q. 15/15

How to remove a substring?

```
/* Use compress function */
proc sql;
   create table data-name as
   select compress(column, ‘replace_with’, ‘search_string’) test
   from some-table;
quit;

/* Replace with nothing ‘’ << empty string */
proc sql;
   create table data-name as
   select compress(column, ‘’, ‘search_string’) test
   from some-table;
quit;
```

Q.

Trick question. Why this code failed?

```
/* someone’s code */
proc sql;
   create table data-name as
   select column-1, column-2
   from some-table
quit;
```

Because he or she forgot a semicolon (;)
Just something small.

```
/* someone's code */
proc sql;
   create table data-name as
   select column-1, column-2
   from some-table;
quit;
```