8.6 L1 password:pd

What vision of SAS do you use?

PC SAS 9.4, SAS studio (the latest version), and SAS Enterprise Guide 7.1

What is library?

A place to save dataset

A dataset?

A file for saving contest

Infile :我要开始读数了

Cards: 我要用Cards来读数，要读取的数字就在Cards之后而不是从其他数据来源

SAS 有numeric and character

区别在于numeric 只能包含数字, 而character可以包含数字+字符

Numeric default length :8bytes(因为数字二进制，所以可以存天文数字)

character default length :8bytes

Dataset/ variable naming: 1, less than 32 bytes 2, only 3 factors （字母，数字，下划线）3, 首位不能是数字

SAS 允许Variable 可以有name 和label （不用再符合以上规定）

Structure of SAS program: 结构排列

1, step data step : create new/update existing dataset

Set 复制

Keep 保留

Where 挑出满足条件的行

Rename 改名

Label 备注

proc step: 分析/处理/展示 data

2,statement datastep stmt

Proc step stmt

\*Global stmt：独立出现的一句话，用于设定

Macro stmt

一句话就是一个statement，每个；代表一个statement结束

3,option global option 包含了global stmt

keyword other dataset option：用括号

考点！：用dataset 时，1，性质变成dataset option 2,要加=

4,Comments

Libname 名字 路径 用来建立一个新的library

8.8

Data SAS data(SAS dataset)

Reading(1,proc import

2.)

Ext.Data(excel,csv,txt..) Temp.Dataset(Work lib)

Perm.Dataset

Example 4 : B

考点！第二种打开xlsx的方法

Libname certxl XLSx ‘…….xlsx’

直接将所有sheet放入SAS里面

Character and numeric valiables

用where 要注意时同种type char to char; num to num

Example 5: missing value(data error)

If cost=”20000”, the result would be 2000

Tips:

Var1 Var2 var= var1+0;( 从numeric 暂时变成character)

‘100’ 100

‘222’ 222

‘333; 333

Char variable

SAS 第一次见到Char variable长度确定了它的Char variable 的长度

所以要在执行前用length 来改变长度

Example 6 : A

‘abc’||’xyz’ ‘abcxyz’ ||用来拼接

Missing data: 面试中常考

Char NUM

‘A’ 1

‘B’ 2

‘ ’ .

Missing value is less then any valid values.

Data quality == Data cleaning

Data issues: 1, missing value

2, outlier eg: age = -20

3, Duplicate

4. Definition error eg: 圆的半径和面积 两个变量计算完之后不同

Missing value:1, find the data owner

2, delete

3, imputation (impute) 3.1 mean

3.2 median

3.3 modeling regression or machine learning

How to deal with missing values?

Drop(‘na’) in python

Quiz 1: shan

怎么算闰年：

除以4，当百年除以400 eg：1900/400

SAS date (1960 JAN 1)

DdMMMYYYYd 才是numeri 的结果

Weekday 1 是周天

14Jul2019 20;48:55 2019.8.8 20:51:30

Diff diff in secs diff in secs

01Jan1960 00:00:00 01Jan1960 00:00:00

Date time Datetime

年月日 时分秒 年月日时分秒

data TEST;

a='13oct2017'd;

a1='13oct2017:13:31:51'dt;

b=today();

b1=datetime();

diff2 = b1-a1;

diff = b-a;

RUN;

Example 7 ：B 因为1960.1.1 是0

Example 8 : A numeric

w.d 4.1 长度的位数为4, 余数的位数为1

Datew. 7. 长度的位数为7

但是SAS 存的数据没变，仅显示格式变化

会四舍五入吗？

面试题

366 ==== 19610101

data test;

a = year(366);

b = month(366);

b1 = put(b,z2.);

c = day(366) ;

c1 = put(c,z2.);

date = a||b1||c1;

run;

Eamxple 9 : A

8.13.2019 lec3&4 lgd(loss given default)

Why role?

Care more the work I do.

Interested in

Need people Who I am have

Why I am here

Contrubte and grow with the team

Strength: technical communication fill in the role they want.

刚刚被面Favour SAS proc step :

Proc Contents: (数据简介)

proc contents data=sashelp.class;run;

如果加VARNUM，第三张表的Variable 会排序

Example 10: D

PROC SORT

proc sort data=sashelp.class out=class;

by sex descending age;

run;

如果Out不写，SAS就会overwrite源文件

If

by sex descending age height;

height是升序，想降序要另外写

proc sort data=cert.overview2 out=overview2\_nodup nodup;

by country;

run;

dedup之后再sort by country

nodup是删掉重复的

nodupkey 会删掉Key variable相同的第二个，以by之后作为Key variable，只会保留第一个

PROC MEANS 算各种统计量的

必考！

proc means data=SASHELP.CLASS sum std mean max maxdec=2;

var Height;

class Sex Age;

run;

Var 是对某一个Variable来计算统计量

Class 是分组

PROC PRINT 打印出来

proc print data= sashelp.class label;

id name;

var sex age height weight;

label sex="Gender: M/F";

format height weight 7.1;

run;

Var 是选取这几个Variables

第一个label，优先打印label，若没有，之后再variable name

第二个Label 是赋予变量备注

Format 两个变量

Id 是将序号取代，以Name 变量为主，并highlight

proc freq 数个数

proc freq data = sashelp.shoes;

tables region;

run;

proc freq data = sashelp.shoes;

tables region\*product;

run;

tables仅仅用于proc freq

Example14： D

Example11:

proc contents data=cert.input11 Varnum;

run;

Example12:

proc sort data=cert.input12 out=results.output12a;

by lastname Firstname descending Age;

run;

CST13652

CST29288

Exmple13:

proc sort data=cert.input12 out=results.output12b ;

by postalcode descending income;

run;

proc sort data=results.output12b nodupkey;

by postalcode ;

run;

proc sort data=result.output12b;

by income;

run;

proc means data=results.output12b mean maxdec=2;

var income;

run;

498

35361

PROC FORMAT

proc format;

value score

low - <60= "Fail"

60- high = "Pass";

run;

proc print data =SASHELP.class;

var weight;

format weight score.;

run;

Value是修饰

SCORE 是定义

定义的时候没. 但是用的时候format 有.

The user-defined format is stored in SAS catalog

Data format

Dataset catalog

Example 15 : C

PROC Transpose

Var是被转置的variable，如果省略了VAR,默认是所有numeric variables

By是被分组的variable

纵变横时

Id 是同一个year里面放置

Prefix 是将新的Variable 加名字

Example 16: 是纵变横！ 体每一列是否变成横C VAR没写

Example 17

By Country

Var Sales

ID Product

Functions

必考

Function 会忽略所有Missing values

Input: 把一个character 永久地读成另一个新numeric

Put : numeric 变成 character

Quiz 2 ：Yan

8.15.2019

面试：

问工资多少：

要体Level！！！与Title无关

不要报钱数，reasonable range

两部分，intro外在，内在compensation

考试最重点 Data manipulation

Index

Text="Australia, US, Denmark";

Pos1=find(Text,'US');

Pos2=find(Text,'US','I');

Pos3=find(Text,'US','i',5);

注意：会包括空格！

放了I 会放弃Case sensitive，所以答案是2

放了I且5，忽略大小写而且从第5格开始

SUBSTR

new\_text=substr(Text,3,4); 从第三位开始取4位stra

Length 不会数 空的trailing blanks

Tranwrd 相当于转换

TRIM 会将空格删掉，只留有用的 如果用TRIM（left()）,就会将前面与后面空的空格都删掉

而Left+ trim 可以用strip代替

SCAN 返回第N个词 delimiters 是指说明哪个符号隔开单词

CATX

SP是seprater， 用来隔开数据的符号

Example18 D

19

Example 20

data results.output20;

set cert.input20;

Major=tranwrd(major,'and ',',');

Secondmajor= scan(major,2,',');

major=scan(major,1,',');

if length (state)=2 then state=upcase(state);

run;

proc freq data=results.output20;

tables major;

run;

proc freq data=results.output20;

tables major secondmajor;

run;

data test;

set results.output20;

where year='Senior' AND state='NC';

run;

proc freq data=results.output20 nlevels;

tables state;

run;

Control Data set flow

面试: Subsetting if 和Where的区别：where的效率要比subsetting if 要高, sustting if 可以出现多次，而where只能出现一次,

1. IF （subsetting if）
2. If.. then

IF condition then action1;

Else action2;

考点IF condition action1;action2; action2会自动执行，不会体if 逻辑。

Example 21 :A

Output

一有output就会将现在的文件输出

如果没有output，SAS会在结尾出产出一个output

Do group:

If condition then do; action1; action2; end;

相当于condition, 做action1,action2

Do loop

Do while (condition) 满足条件时继续执行

Do Until (condition) 不满足条件时继续执行

Example 23 :D

Example19:

data results.input19;

lenth name $18;

set cert.input19;

fIRSTNAME = TRANWRD(FirstName,'William','Will');

name=CATX(',',lastname,firstname);

len=length(name);

IDnumber = substr(CustomerID,4,5)+0;

if find(name,'Z','i') >0 then Zflag=1;

else Zflag=0;

run;

proc means data=results.input19 sum;

var len IDnumber Zflag;

run;

proc means data = results.input19(where=(firstname='Will'));

var age;

run;

8.20 lec5-6 password:ead

工作中:proc freq 要很好的表达出来

考点就在ods 的两项，要成对出现的，ods开头的，或记住ODS的语法

如果是改变存另外一种文件类型，一定要改三处

Example24: D

Example25:D

Example26: C SAS有包容性

Example27:

ods pdf file="/home/u40030426/OUTPUT/output27.pdf";

title;

footnote;

proc print data=cert.input27;

run;

ods pdf close;

ods rtf file='/home/u40030426/OUTPUT/output27.rtf';

title;

footnote;

proc print data=cert.input27;

run;

ods rtf close;

176.6

675.6

Combine Data sets

考点：one to one reading 无脑的横向拼接dataset 名字一样的，就第二个覆盖第一个，以最短的dataset为主 难点在于

Set 文件1；

Set 文件2；

Data Concatenation 纵向拼接 首尾相连 难点在于：Set 文件1 文件2；加By就可以sort

面试：

纵向拼接的关键是：这两个文件本身要先排序，如果唔系没办法执行。

Merge 横向拼接

Merge file1 file2;

By key variable;

Merge/join inner join, left join, right join, full join,

data merge\_2;

merge work.a(in=ind\_1) work.b(in=ind\_2); in是建立temp variables，意思就是体有没Contributed temp variable只能取0或1, 用了if就是为了选取适合这个条件的

有个超级大前提！ 这两个dataset要先排好序！而且保证不是many to many（即没重复）

merge work.a(in=ind\_1) work.b(in=ind\_2);

by num;

if ind\_1=1; \*/left join

/\* if ind\_2=1; \*/right join

/\* if ind\_1=1 and ind\_2=1; \*/inner join

/\* if ind\_1=1 or ind\_2=1; \*/full join sas default

run;

Example28：B

Example29:

data results.output29;

set cert.input29a;

set cert.input29b;

totalPrice=qty\*price;

run;

proc means data=results.output29(where=(product="Screwdriver")) sum;

var totalPrice;

run;

proc means data=results.output29 sum;

var totalPrice;

where product="Screwdriver";

run;

proc means data=results.output29(where=(product="Tape Measure")) mean;

var totalPrice;

run;

Example22:

就是因为用了OUTPUT，就不会输出ouput以下的内容

所以用Do 来执行两个Actions

Quiz1

1,记得要加XLSX

2xlsx

3, 2个 首位只能是字母或下划线

4,c

5,D

8.22 考试：左边是介绍，右面是操作界面。

一定要follow 左面的步骤

一定要先设libname

libname cert "C:\cert\input";

libname results "C:\cert\output";

最后半个小时一定要转选择填空

Quiz 3 mitch

Retain & sum 做累加

data end;

retain sum 0; 初始值为0，而且让SAS记住Sum之前的值，不然会变成missing value

set test;

/\* sum=i+sum; \*/

sum=sum(i,sum);

run;

do loop 不熟 因为用加号，missing value + 5050 最后是missing 所以要用sum function

加法 ： +

Sum function

Sum stmt 累加（与加法不一样）而且自带ignore missing value的功能

Sum statement syntax:

Variable + expression;

Eg:

data end2;

set test;

sum + i;

run;

Example 30: A D选项因为用了加号累加，所以会有missing value

累加会有一道大题

Sas Automatic variables

Where 不能，而IF 可以与automatic variable 一起用

Eg； DATA…; first.id last.id

BY id year; first.year last.year

RUN…;

BY 如果有两个变量的话，要体一个pair

如果想用By, 要先排序（同理 Merge）

Example 31 :A each 后面是谁就选谁

当见到

By

If

Sum statement

If

就是分组累加

If first.insert code then payroll=0 当first.insert code 等于1时，payroll =0

If last.insert code. 当last.insert code 等于1

Example32: C

SAS MACRO language

面试工作

Basie 考试只考第一个

%let

Resolve: 1,&

2, 单双引号： 只有双引号才能正确resolve，单引号不能

%let i=5; \*what if I want resolve "i" in "ith";

proc print data=cert.overview(obs=&i firstobs=&i);

title "&i.th Record in work.overview"; \*resolved as 10th;

title2 "&ith Record in work.overview"; \*can not resolve, since ith is not a macro variable;

run;

1. 要加.不然会没办法分辨

%put &data &I;

会出现你设置Macro的variables

Example34: A

Example35: &amount

Example36： A

Example37:

%LET TARGET = 8;

所有的Example都要自己做

Quiz2

1. 如果不写Var，SAS会自动将Num transpose

B

2, C

3，D

4, 2

5, C

6，B（区分大小写）

7，B

Quiz3：

Q3：C

Q4：B

8.27.2019

SAS Advanced

Advanced 延期到10月31号改版

数据从哪来？

Extern.Data(Csv. Excel,txt) Relational Database (Oracle,Tera data,DB2)

Proc inmport SQL

Libname

Data

在SAS 中的PROC SQL 是interactive，所以可以一行一行来执行，但是要用Quit来结尾

最好去用SQL去做SAS

Select 1, as 1, 改名，eg: select year as fiscal year 2起名 select year\*0.1 as number

2,label or format eg: select hiredate label=’Date of Hire’

select hiredata format=date9.

3,Distinct eg: proc sort …. Nodup;

Proc sort… nodupkey;

Distinct与nodupkey是一样的

4，\*

Example1 C

Example2 C 因为没有写By，

Where 1,Calculated 当你选取新的variable,而不是原有的variable时，要加上calculated

Eg: where calculated variable…

即使是select 的时候，如果想建一个新的变量，

Example 3 D is not missing 适用于num和char 变量 仅在Sas 用

Orderby 1,symax

Proc sort data=…;

By var1 descending var2;

Order by var1, var2 desc;

Order by gourp by 可以用1，2，3，4来表达select 的变量

Group by 你可以用summary function，如果没用summary function 就当成order by

Having By 是使用于Group level 来选取符合条件的

Example 4, D 因为有Group by所以才会出现2个

Where和having的区别：一个是行的选择 一个是组

where不能跟一个variable的summary function，having 可以

where 先执行 having后执行

where 加新建的列要加calculated 而having可以不加

EXAMPLE 5 B 因为where后面不能加单个的variable summary function

Inner join

第一种 from.. where…. 逗号加where 的inner join，不加where就是cartesian product

第二种 inner join… on

Example 7 C

如果是inner join 是两行

2001 3 500 2001 4 300

2001 4 400 2001 4 300

Left join

2001 3 500 2001 4 300

2001 4 400 2001 4 300

2003 1 350 . . .

Right join

2001 3 500 2001 4 300

2001 4 400 2001 4 300

。 。 。 2002 1 600

Full join

2001 3 500 2001 4 300

2001 4 400 2001 4 300

。 。 。 2002 1 600

2003 1 350 . . .

因为select所以

SAS的outer join 不是SQL中的full outer join

SQL和SAS的merge结果会不一样

1，Full join

2, 两个Key variable 不相同

Coalesce function 将两列合并到一列

Example6: C

BD都是inner join

8.29.2019

Vertically combine 考点

ID year ID year

====》

ID Amt

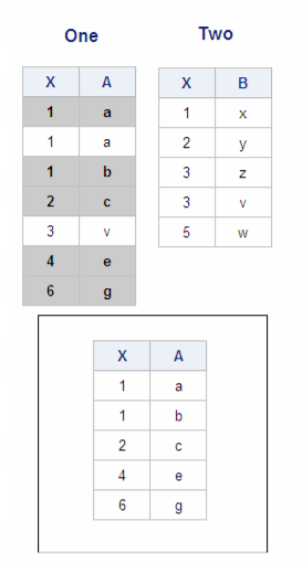
用SQL纵向合并时，只会体位置，而不会体变量。

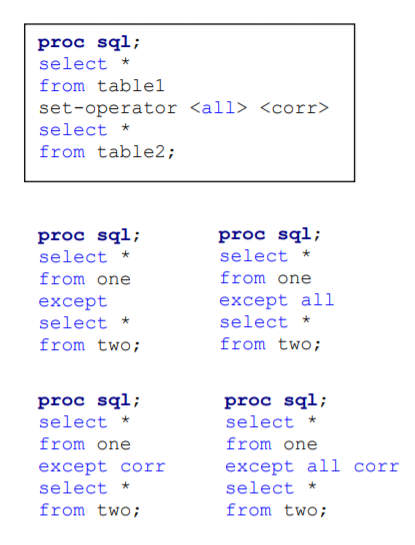
所以Amt会被吞掉

但是如果用Set Ds\_1 Ds\_2;

ID YEAR AMT

用SQL纵向合并时，过程是：先在内部做dedup，再选符合条件的

在Note 1 SQL for student 的15页



右上方的加了ALL之后就不做Dedup了。再合并。

会比唔加ALL多了一行 1 a

左下角corr corresponding X列就是corr（两个表共有的variable）

加了corr就只负责corr 的列 其他都不管

先dedup是变成

12346

1235

所以答案是

4

6

右下角

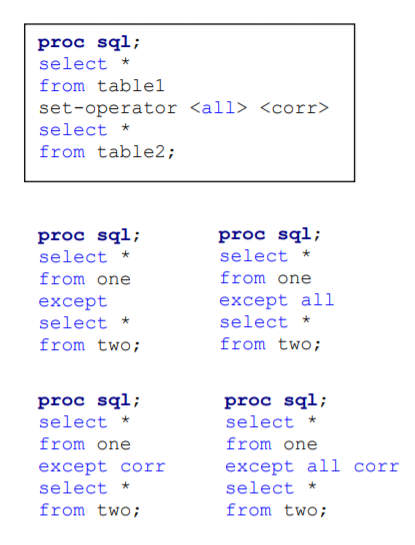
先体到corr，只负责corr的列

All就不做dedup

左边的Overlap只会带走一个1

答案是

1146



当用intersect时

左上角 答案时

3v

右上角 还是3v

左下角 1 2 3

右下角 1 2 3

当用union时 先做纵向合并 再做dedup

左上角

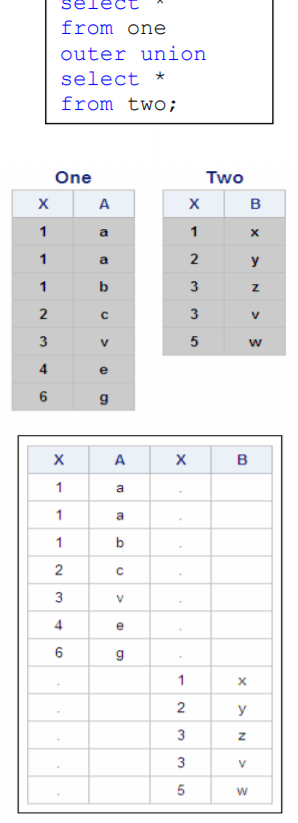
右上角 全部直接合并

左下角 只看x 先纵向合并再dedup 123456

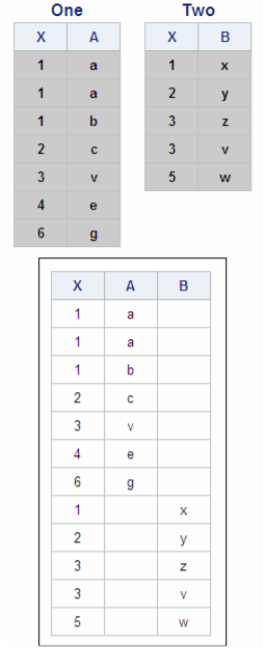
右下角 直接合并

Outerunion

1，不做dedup

2，两者不会合并

Outerunion因为不做dedup所以只会加corr



Data step 纵向合并等同于outerunion corr

纵向合并：

1，set ds\_1 ds\_2;

2, SQL

3 proc append

Example 8: C 因为有dup 所以肯定OUTERUNION 因为outernuion不做dedup

EXAMPLE10: B

Example 11: B

Example 9 : A BC的选项是inner join D是会有三列

Nested query——sub query

有两类 uncorrelated correlated（工作中能不写就不写）

如何判断，遮住主query 如果能单例执行，就是uncorrelated

考点

执行顺序 uncorrelated只运行一次，且先于主query 执行

Exists 判断subquery有没有value会被返回，ture or false

如果是correlated query会执行N次，因为会根据主query来执行 耗时超长

用where进行比较时，最好避免一个数与一堆数比较 eg:

Where dateofbirth < any(subquery…)

Where dateofbirth < all(subquery…)

Example 12: C 等于号选择in，不等号可以选择any or all

Proc sql;

Select max(salary) from test

Where salary < (select max(salary)from test);

Quit;

而在from的sub query 就叫in-line views

但是inline views 不能有order by

Example13:最难 最重要

A是可以复刻出来的 A中的salary是表中的additional vairbale(select后接组名和summary function)

B corr 如何执行？

C uncorr 选择C 因为subquery会返回3个数，

D From table A table B inner Join

Example14 :A

Example 15:B

Index

Work.employee

ID Hire\_yr

001 2001

002 2001

003 2004

004 2007

将Hire\_yr 作为index

2001 1，2

2004 3

2007 4

如何在SQL建立Index

Example16:A

Example17:D

Example18:C

SQL如何练习？

EXAMPLE

找工作？

W3school

面试题？

9.3.2019

Lec 2

MACRO 面试超多

‘Automation improve effiectivce’就是Macro

1，Macro variable

1, how to create

1, %let Macro stmt (简单，但是一次只能建立一个macro variable)

2,call symput datastep stmt (可以建立多个macro variable)

3, proc sql

2, resolve 使用

1,& 来使用

2，单引号 不可以resolve

双引号 可以resolve

3， ‘&i.th’ ’&ith” 前者可以选到第i个

4. %put 会报出macro variables

5. &&&

EXAMPLE19 C A错在没双引号

EXAMPLE20 C Birthday is num where 中 等号左右一定要相同格式

面试中建立macro variable

call symput

data \_null\_;

set overview;

call symput(Record\_No,gdp); 放了两个不同的列，前面的macro variable的名字，后面的macro variable的value

run;

%put \_user\_;

是所有user设定的信息

data \_null\_;

set overview;

If \_n\_ =1 then call symput(Record\_No,gdp); 只建立第一行的macro variable

第二个变性

data \_null\_;

set overview;

If \_n\_ =1 then call symput(“Record\_No”,gdp); 引号内的内容将是macro variable 且只建立第一行

第三次变性

data \_null\_;

set overview;

Call symput(‘line”||compress(\_n\_),put(gdp,dollar10.1)); 工作中常用

Compress 临时将num变成char

同样的，逗号前面是macro name，逗号后面是num 变成char

PROC SQL; 工作常用

select count(\*) into: n from work.overview;

select into 将数据放入n 的macro variable

select distinct country into: country\_list separated by "," from work.overview;

当返回多个value的时候，一定要用separated by，不加就只返回第一个value

quit;

Example 21: C

Consecutive &

Rule 1: SAS read left to right • Rule

2: &&→& • Rule

3: In each read, SAS only resolve each & once • Rule

4: SAS will read again while there is & unresolved

%let a=alpha; %let b=beta; %let abeta=pair;

%put &&a&b;

所以变成&a&b

再变成&abeta

之后再变pair

%let attri=name;

%let name=shan;

%let shan=chinese;

%put &name is a &&&&name; &&name &name shan

%put &name is a &&&&&name; &&shan &shan Chinese

%put &name is a &&&&&&name; &&&name &shan Chinese

Example23

%put &&&Root&Suffix; &name2 Clothes

Macro variable 名字可以大小写

Example 22 B

Macro function 只作用在macro variable

%let a= Shan's favorite mathematicians;

第一个问题，空格不会表达。第二个 ‘， 会一直等第二个引号

%let b=&a are; Newton&Gauss;

第一个问题，第一个； SAS会认为结束了

第二个问题，& SAS会认为resolve Gauss

%STR会解决引号，分号，空格（但是要加%）

%NRSTR 与%STR 同时不会resolve variables

%let a=%str( Shan%'s favorite mathematicians);

%let b=%str(&a %nrstr(are; Newton&Gauss; &a)); nrstr内面的&a是用来区分

%put &a &b;

1，%str…

2%let a=1+1;

%put &a

会是1+1

%eval 只允许整数运算，而且只输出结果的整数部分，而且不会四舍五入（round up）

%sysevalf 可以进行小数运算（常数而且不能用function 例如要开平方只能是\*\*0.5的写法，）

%sysfunc 可以使用function且可以选取格式

Example24 D

Example25 B

3， datasetp func macro version

Num Char Macro variable

7 ‘ABC’ ABC (就是text value)

Example26 D 没有引号因为where 要两边格式相同 C没有百分号，要作用于macro vairable

MACROS 模板！！

程序模板类似于雅思的作文模板

%macro print(data,var=); 前者是positional而且必须先于keyword 后者是keyword，带等号 而工作中更多是只定义keyword，第一，可以不用记顺序。第二，可以设置默认值

proc print data=&data noobs label;

var &var;

run;

%mend;

%print(work.overview,var=country year gdp); 调用的时候于上面的顺序是一样的

Example27 C value分大小写

Example28 B

Example29 A 第一句调用是错的，所以不会运行。第二句会根据默认值的运行

调用的时候可以不加分号，因为其实自动填充回原句，而原句有分号即可

Example30 D

9.5.2019

MACRO related Global Options 设定，主要是debug的作用

考试会出一个log，遮住，然后比你体其他的过程，之后问你是哪个Global options

SYMBOLGEN 考点

会解释你的Macro variable resolves 的来源，记住log中会有resolves to

MPRINT

会将最近被执行的step 过程

MLOGIC

会提供每一个逻辑判断的结果

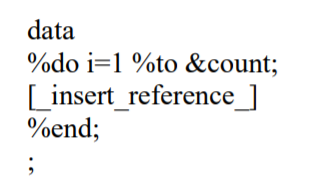
Example31 C Mprint会将所有都打印出来，而不是单个parameter

Conditional Processing and Iteration

不能用在Open code，只能用在macros definition(%macro………到%Mend)

Example32(最好的一题) B

Data \_null\_ 只执行，不生成dataset

 最好变成 data not serious；

Work.uniques

Status

Not

Serious

set uniques end=Lastobs; set的一个option， 会添加一个列lastobs，value取值所有行是0，

最后一行是1

call symputx('Status'||left(\_n\_),Status);

会生成两个macro vairables Status1（not） 和Status 2（serious）

if Lastobs then call symputx('Count',\_n\_); 因为lastobs 是1时，n是2

data %do i=1 %to &count; &count已经是2

[\_insert\_reference\_] &&status&i； 如果&i是1，就&status1，所以答案是not

如果&i是2，就serious

%end; ;

所以Data not serious

Select

When 就相当于if…then do

Scope of macro variables – Global and Local

Macro variable的分类

Global symbol table （用来存macro variable，仅持续到关闭SAS）

MV1 V1

MV2 V2

Local symbol table(也存macro variable，建立时间是每次Macro调用的开始，到调用结束。)

MV1 V1

MV2 V2

Local

• parameters in a macro definition

• a %LET statement within a macro definition

• a DATA step that contains a SYMPUT routine within a macro definition

• a DATA step that contains a SYMPUTX routine within a macro definition

• a SELECT statement that contains an INTO clause in PROC SQL within a macro definition

• a %LOCAL statement.

Note: The SYMPUT routine can create a local macro variable if a local symbol table already exists. A local symbol table is created when a macro that includes a parameter list is called or when a request is made to create a local variable during macro execution.

并不是所有的任意的Macro调用，local marco都会被建立

第一：一个marco包括parameter list

第二：macro里面带有建立local macro table的请求

Example33: C

%let Mv=shoes; Global

%macro PRODUCT(Mv=bicycles); MV shoes

Local

刚调用MV tents

%let Mv=clothes; 执行时 MV clothes

%mend; 之后执行时 没有local table了

Example34 B

Local 第一个put的时候，work.new sashelp.sas

Newname work.new

Setname sashelp.class

之后就消失了。

所以newname setname原话返回出来

Writing the macro variables into the SAS Log

\_ALL\_ 所有MACRO VARIABLES

\_AUTOMATIC\_ 只输出sas自动建立的， 所以automatic +user =all

\_GLOBAL\_ 所有Global macro variables global+ local =all

\_LOCAL\_: can only display when defined within macro 所有local macro variables

\_USER\_

新题

%let a=4579;

%let b=12345;

%put ….. 答案是 \_global\_

想看到global a 4579

Global b 12345

SAS Automatic Macro Variables

SYSDATE 是你打开SAS的日期，而不是当天

Example35 D

Example36 B 等号两边要相同格式

SYSLAST the name of the most recently created SAS data set

SYSERR

%symdel var; 在global中删去macro variables

Example37 D

Quiz题（SQL 和MACRO）

找工作：

1，拿面试

1，networking

Linkedin responsible rate？

校友 position(team member)

Job fair/Infor section --------简短的自我介绍，问自己的问题：第一，有没有opening，有多少，什么时候，哪个方向，你们要招什么样的人，能不能看一眼简历，能否留一下简历。

2，海投（越早越好） 学校，

Linkedin，indeed technical skills

简历的summary of qualication 就是要对上jd的requirment

从工作的逻辑来

2，拿下面试

9.10.2019 Lec3

有效率：

SAS 几千万行

Excel 几万行

Sas几个版本

SAS studio

PS sas 9.4

Sas ENTERPRISE GUIDE 7.1

SAS9.4 ,R, PYTHON软件，数据都存在电脑上，所以配置，硬盘都会影响运算效率

SASEG 就是解决这个问题，将服务器装一个SASEG

将所有数据放在服务器，服务器有多核，供多人同时工作。

然后所有运算在服务器上运行，电脑写code，投射给服务器，在服务器处理数据，存数据

银行中都有一个数据库，Database上面装系统（Oracle/teradata），先拖数据到自己电脑或者是拖到服务器（用SQL）

Computing resource

1，CPU

2, Memory 内存 （临时储存）能与CPU交换数据很快

3，hard disk 硬盘（永久储存）

4，I/O input/output

Hard Disk Memory Memory Hard Disk

DS\_1 🡺 Buffer ==》 CPU/PDV ==》 Buffer ====🡺 DS\_2

➢ Improvement in I/O can come at the cost of increased memory consumption.

节省I/O会增加memory的使用

➢ When you create a SAS data set using a DATA step, the following actions occur:

• SAS copies a page of data from the input data set to a buffer in memory.

• One observation at a time is loaded from the buffer into the program data vector.

• Each observation is written from the PDV to an output buffer.

• The contents of the output buffer are written to the disk when the buffer is full.

Page Size == Buffer SIZE

• It is the unit of data transfer between the storage device and memory.

• It is fixed in size when the data set is created, either to a default value or to a user-specified value.

A larger page size can reduce execution time by reducing the number of times SAS has to read from or write to the storage medium. However, the improvement in execution time comes at the cost of increased memory consumption.

How to improve I/O?

1,bufsize = bufsize变大，pagesize变大所以会变快

2，bufnum= 增加buffer在memory的数量，所以会变快

Increasing the value of either option increases the amount of data that can be transferred in one I/O operation. BUFSIZE BUFNO Bytes Transferred in One I/O Operation

6144 2 12,288

6144 10 61,440

30,720 2 61,440

30,720 10 307,200

就是buffsize\* buffnum

Example38 D

SASFILE statement

就是告诉SAS将某一个dataset 一直保存在memory中，阻止memory清空（因为每次执行后，memory都会清空）

考点：SASfile 的功能：

1，decrease I/O

2, increase memory

3, decrease CPU

SASFILE SAS-data-file OPEN | LOAD | CLOSE;

Open 和 load的区别：

Load马上进memory并且hold住

Open是等等后再进memory

➢ The SASFILE statement opens a SAS data file and allocates enough buffers to hold the entire file in memory. 自动弄进去hold在里面

It is important to note that I/O processing is reduced only if there is sufficient real memory.

至少有足够的内存

Example39 D

I/O 下降，Memory 一定会上升

SAS benchmark option

STIMER Specifies that CPU time and real-time statistics are to be tracked and written to the SAS log throughout the SAS session.

MEMRPT Not available

FULLSTIMER Specifies that all available resource usage statistics are to be tracked and written to the SAS log throughout the SAS session.

STATS Not available

Example40 B

Example41 考点 A

Example42 C

Example43 B

节省Hard disk

Compressed file 压缩

OPTIONS COMPRESS= NO | YES | CHAR | BINARY;

Yes 和 char是同一种压缩算法 RLE 主要用char 带有repeated balnks或者num 带有很多zero

Binary用另一种压缩算法 RDC 适合非常大的observations 中到大型的char带有特定pattern的

Example44 B

REUSE=

只能用于压缩的文件

Example45 B

到底要不要压缩？

大，repeated, missing values,

压缩文件会需要更多的CPU 你想节省HARD DISK，CPU就会增加能耗

最后一个节省硬盘空间

DATA step views

Data sas data dataset temp

Perm

Datastep views(不存数据，只存instructions eg:指令)

节省hard disk 浪费了cpu 因为每次都要先处理数据

所以如果有变化的dataset（live dataset）,data step views更好用因为每次都用最新的数据来处理

Example 46 D

Note: If you specify additional data files in the DATA statement, SAS creates these data files when the view is processed in a subsequent DATA or PROC step. Therefore, you need to reference the data view before you attempt to reference the data files in later steps.

建了两个一个是view，另一个就是additional data file

View与其他混在一起，只有views执行后，剩下的dataset才会被建立

相当于View 是先体，不做。做的时候再按顺序

Example47 B

Example48 B

The DESCRIBE Statement 查view中的statements

Referencing a Data View Multiple Times in One Program

还不如存成一个temp sas data file

今天跟明天占30少一点，但是容易出新题

9.12.2019 最后一节课

Index 提速查找过程

• Yield faster access to small subsets of observations for WHERE processing

Small subsets 是占总数3%以内，3%到33%，有可能使用index，33%以上不用index

不用index时，SAS就每个去数

Index 包括value 和identifier pair

Index 分类 考点

Simple （自动起名） composite （需要起一个名字）

一个variable 多个variable

建立index 的三种方法

In data step

基于一个variable,但是建了两个，（先check empid是否unique，如果是，再建立index）

data simple2 (index=(division empid/unique));

An index is not used in these circumstances:

• With a subsetting IF statement in a DATA step

• With particular WHERE expressions

• If SAS determines it is more efficient to read the data sequentially

基于两个variable

data composite (index=(Empdiv=(division empid)));

如何建立，删除index：

In proc step

proc datasets library=sasuser nolist;

modify sale2000;

index create origin;

quit;

proc datasets library=sasuser nolist;

modify sale2000;

index delete origin;

index create flightid;

index create Fromto=(origin dest);

quit;

In sql

proc sql;

create index origin on sasuser.sale2000(origin);

quit;

proc sql;

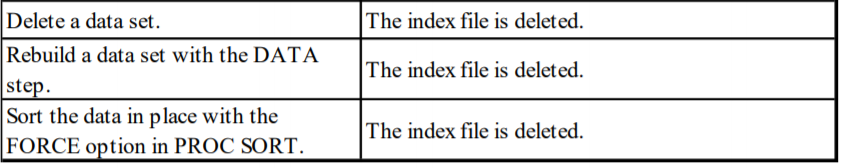
drop index origin from sasuser.sale2000;

create index Tofrom

on sasuser.sale2000(origin, dest);

quit;

同一个library，有一个或多个index就会存在index file



用了proc sort就会自动删掉index

Proc copy， index 也会被copy

Proc copy + move option 就是剪切 index moved

Rename 1, dataset name 还想保留index, 就用proc datasets + change stmt

(只要改名都会将Index删去)

2, variable name 都会删去index

1, rename stmt

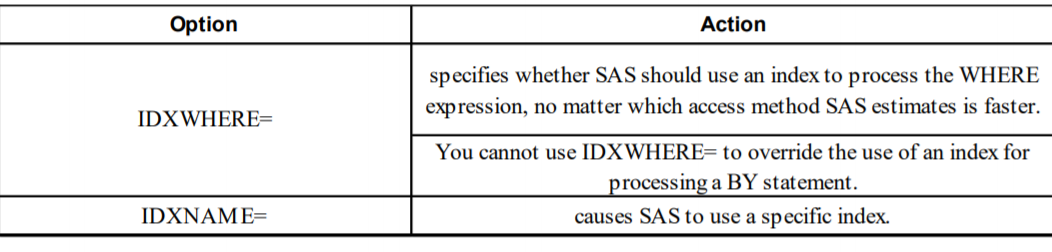
2, datastep; rename dataset option

还想保留index, 就用proc datasets +rename stmt

Example49 A

Example 50 D

Example 51 A 因为有一个index 在id, 所以index是升序的，所以可以执行



Idx name 后面加 强行要你使用，

但是idxwhere idxname不能同时用

Multidimensional array 考点 面试题

Array 在SAS只用来variable 起别名，必须全部都是同类型的

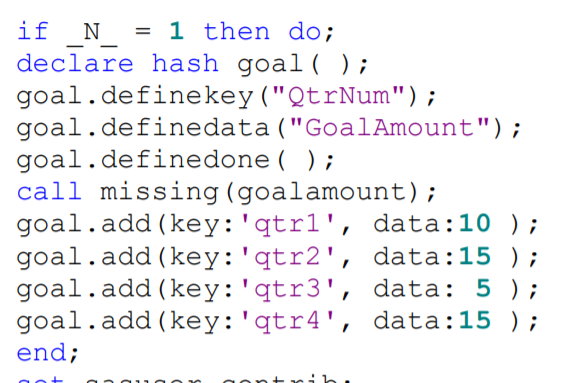
Example52 A

考题：array Yr(1:2,5:7) 相当于 yr(2,5)

Hash Object(只考一道题)

有很多可以功能可以代替

专门用来lookup ，专门放在memory，用完就弃，但是总体时间很短



考点在于

Goal.add(key: Var1, data: Var2);

Example 53 B

Using Custom Formats （只考一个题）

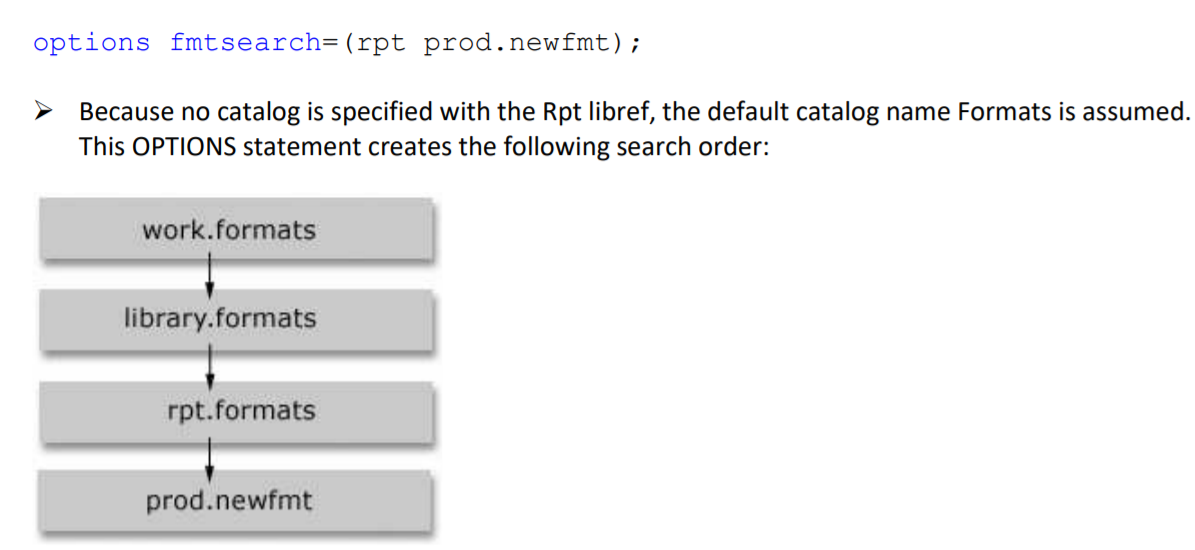
When a format is referenced, SAS automatically looks through the following libraries in this order:

1. Work.Formats

2. Library.Formats

Example 54 A

OPTIONS FMTSEARCH= (catalog-1 catalog-2...catalog-n);

就是用来自己定义搜索的过程

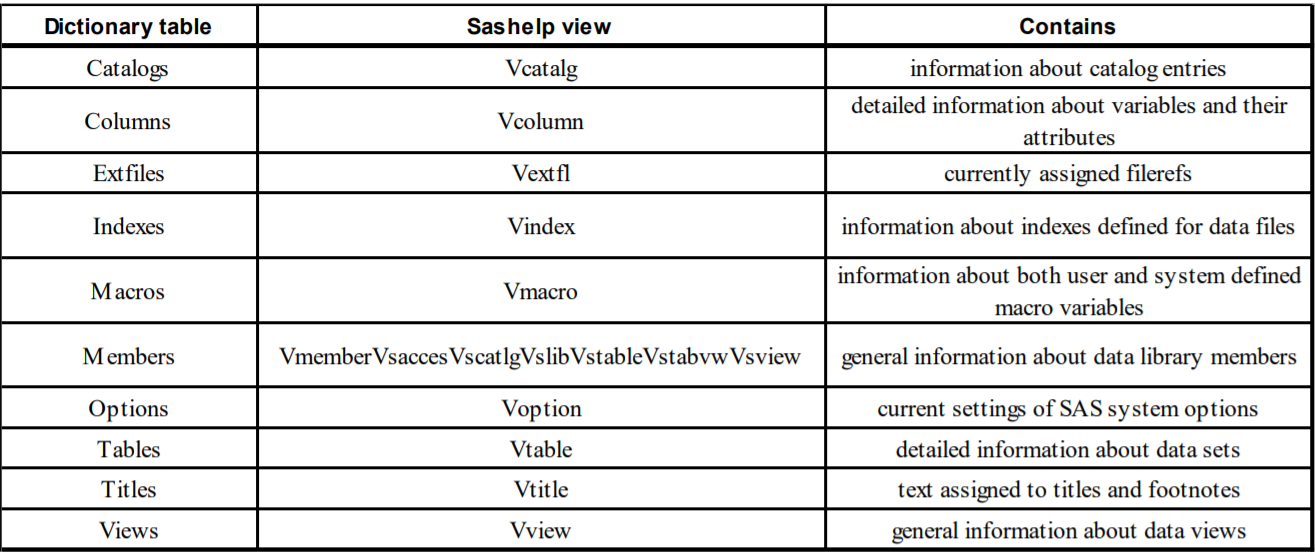
Example 55 A

Dictionary Tables

Example 56 B

created each time they are referenced in a SAS program

• updated automatically



Columns and Titles 有机会被考到

‘ORION‘ 包括了很多个datasets

从哪个dictionary table有一个dataset 可以查到以num结尾的variable的名字

Dictionary columns 中的模糊查找可以

证书考完之后，应该怎么做？

有个小任务，一定要做 project experience

Advance是SQL 和MACRO

Quiz 和EXAMPLE

4个Quiz