**Ch 3**

**Level 1**

**4.**

a.

proc print data=orion.country;

run;

| **Obs** | **Country** | **Country\_Name** | **Population** | **Country\_ID** | **Continent\_ID** | **Country\_FormerName** |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | AU | Australia | 20,000,000 | 160 | 96 |  |
| **2** | CA | Canada | . | 260 | 91 |  |
| **3** | DE | Germany | 80,000,000 | 394 | 93 | East/West Germany |
| **4** | IL | Israel | 5,000,000 | 475 | 95 |  |
| **5** | TR | Turkey | 70,000,000 | 905 | 95 |  |
| **6** | US | United States | 280,000,000 | 926 | 91 |  |
| **7** | ZA | South Africa | 43,000,000 | 801 | 94 |  |

How many observations are in the **orion.country** data set? 7 observations /\*reading from the log\*/

How many variables are in the **orion.country** data set? 6 variables /\*reading from the log\*/

What is the name of the last country in the data set? South Africa /\*seen above\*/

b. Submit a PROC CONTENTS step to generate a list of all members in the **orion** library. What is the name of the last member listed? US\_SUPPLIERS /\*reading from the results\*/

proc contents data=orion.\_all\_ nods; /\*this is how we get the list of all members of the orion library-all\_nods\*/

run;

| **Directory** | |
| --- | --- |
| **Libref** | ORION |
| **Engine** | V9 |
| **Physical Name** | /folders/myfolders/Data\_One/Data\_One |
| **Filename** | /folders/myfolders/Data\_One/Data\_One |
| **Inode Number** | 445 |
| **Access Permission** | rwxrwx--- |
| **Owner Name** | root |
| **File Size** | 80KB |
| **File Size (bytes)** | 81920 |

| **#** | **Name** | **Member Type** | **File Size** | **Last Modified** |
| --- | --- | --- | --- | --- |
| **1** | CHARITIES | DATA | 128KB | 11/17/2015 03:49:52 |
| **2** | CONSULTANTS | DATA | 128KB | 11/17/2015 03:49:52 |
| **3** | COUNTRY | DATA | 192KB | 11/17/2015 04:25:12 |
|  | COUNTRY | INDEX | 32KB | 11/17/2015 04:25:12 |
| **4** | CUSTOMER | DATA | 128KB | 11/17/2015 03:49:52 |
| **5** | CUSTOMER\_DIM | DATA | 128KB | 11/17/2015 03:49:52 |
| **6** | CUSTOMER\_TYPE | DATA | 192KB | 11/17/2015 03:49:54 |
|  | CUSTOMER\_TYPE | INDEX | 24KB | 11/17/2015 03:49:54 |
| **7** | EMPLOYEE\_ADDRESSES | DATA | 192KB | 11/17/2015 03:49:52 |
| **8** | EMPLOYEE\_DONATIONS | DATA | 128KB | 11/17/2015 03:49:52 |
| **9** | EMPLOYEE\_PAYROLL | DATA | 128KB | 11/17/2015 03:49:52 |
| **10** | LOOKUP\_COUNTRY | DATA | 128KB | 11/17/2015 03:49:52 |
| **11** | MNTH7\_2011 | DATA | 128KB | 11/17/2015 03:49:52 |
| **12** | MNTH8\_2011 | DATA | 128KB | 11/17/2015 03:49:52 |
| **13** | MNTH9\_2011 | DATA | 128KB | 11/17/2015 03:49:52 |
| **14** | NONSALES | DATA | 128KB | 11/17/2015 03:49:52 |
| **15** | NONSALES2 | DATA | 128KB | 11/17/2015 03:49:53 |
| **16** | OLDBUDGET | DATA | 128KB | 11/17/2015 03:49:53 |
| **17** | ORDERS | DATA | 128KB | 11/17/2015 03:49:53 |
| **18** | ORDER\_FACT | DATA | 128KB | 11/17/2015 03:49:53 |
| **19** | ORDER\_ITEM | DATA | 128KB | 11/17/2015 03:49:53 |
| **20** | ORGANIZATION\_DIM | DATA | 320KB | 11/17/2015 03:49:54 |
|  | ORGANIZATION\_DIM | INDEX | 40KB | 11/17/2015 03:49:54 |
| **21** | PRICE\_CURRENT | DATA | 128KB | 11/17/2015 03:49:53 |
| **22** | PRICE\_NEW | DATA | 128KB | 11/17/2015 03:49:53 |
| **23** | PRODUCT\_DIM | DATA | 256KB | 11/17/2015 03:49:54 |
|  | PRODUCT\_DIM | INDEX | 64KB | 11/17/2015 03:49:54 |
| **24** | PRODUCT\_LEVEL | DATA | 192KB | 11/17/2015 03:49:54 |
|  | PRODUCT\_LEVEL | INDEX | 24KB | 11/17/2015 03:49:54 |
| **25** | PRODUCT\_LIST | DATA | 192KB | 11/17/2015 03:49:54 |
|  | PRODUCT\_LIST | INDEX | 48KB | 11/17/2015 03:49:54 |
| **26** | QTR1\_2007 | DATA | 128KB | 11/17/2015 03:49:53 |
| **27** | QTR1\_2011 | DATA | 128KB | 11/17/2015 03:49:53 |
| **28** | QTR2\_2007 | DATA | 128KB | 11/17/2015 03:49:53 |
| **29** | QTR2\_2011 | DATA | 128KB | 11/17/2015 03:49:53 |
| **30** | SALES | DATA | 128KB | 11/17/2015 03:49:53 |
| **31** | SALESQUIZ | DATA | 128KB | 11/17/2015 03:49:53 |
| **32** | SHOES | DATA | 128KB | 11/17/2015 03:49:54 |
| **33** | SHOES\_ECLIPSE | DATA | 128KB | 11/17/2015 03:49:54 |
| **34** | SHOES\_TRACKER | DATA | 128KB | 11/17/2015 03:49:54 |
| **35** | STAFF | DATA | 192KB | 11/17/2015 03:49:54 |
|  | STAFF | INDEX | 64KB | 11/17/2015 03:49:54 |
| **36** | SUPPLIER | DATA | 192KB | 11/17/2015 03:49:54 |
|  | SUPPLIER | INDEX | 24KB | 11/17/2015 03:49:54 |
| **37** | US\_SUPPLIERS | DATA | 128KB | 11/17/2015 03:49:54 |

**Level 2**

**5. Viewing General Data Set Properties**

b. What sort of information is stored for this data set? This dataset stores employee information like Start Date, End Date, ID, Salary, Gender, etc. There are 10 variables. /\*Read from the log\*/

**Challenge**

**6. SAS Autoexec File**

What is the name of the file? The default name of the file is AUTOEXEC.SAS

What is its purpose? Contains statements that are executed immediately after opening SAS, before any user input

How is it created? Using a SAS Text Editor, or if you must another ASCII editor, but not Word. Then just save as, or go through the AUTOEXEC file to name it something other than AUTOEXEC so that SAS knows where to find it.

How could this be useful in a SAS session? Useful if you always want to work on a specific file, or can open commonly worked on files.

**Ch 4**

**4.1**

**Level 1**

1. **Displaying orion.order\_fact with the PRINT Proceudre**
   1. proc print data=orion.order\_fact;
   2. Add a SUM statement to display the sum of **Total\_Retail\_Price**.

proc print data=orion.order\_fact;

sum Total\_Retail\_Price;

run;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. **613** | 240800100042 | 3 | $760.80 | $105.30 | . |
| **614** | 240500200016 | 3 | $95.10 | $14.50 | . |
| **615** | 240500200122 | 2 | $48.20 | $11.50 | . |
| **616** | 240700200018 | 4 | $75.20 | $10.30 | . |
| **617** | 220101400130 | 2 | $33.80 | $5.70 | . |
|  |  |  | **$100,077.46** |  |  |

* 1. Add a WHERE statement to select only the observations with **Total\_Retail\_Price** more than 500.

proc print data=orion.order\_fact;

sum Total\_Retail\_Price;

where Total\_Retail\_Price>500;

run;

| **Obs** | **Customer\_ID** | **Employee\_ID** | **Street\_ID** | **Order\_Date** | **Delivery\_Date** | **Order\_ID** | **Order\_Type** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **22** | 79 | 121045 | 9260101874 | 18APR2007 | 18APR2007 | 1230591684 | 1 |
| **41** | 88 | 99999999 | 9260100179 | 25MAY2007 | 28MAY2007 | 1230793366 | 2 |
| **54** | 2806 | 99999999 | 8010100089 | 04JUL2007 | 08JUL2007 | 1231014780 | 2 |
| **61** | 61 | 99999999 | 3940108887 | 20JUL2007 | 25JUL2007 | 1231094514 | 2 |
| **72** | 70187 | 99999999 | 2600100035 | 13AUG2007 | 18AUG2007 | 1231227910 | 2 |
| **120** | 19 | 99999999 | 3940106547 | 26DEC2007 | 30DEC2007 | 1231976710 | 3 |
| **170** | 34 | 121021 | 9260111379 | 14MAY2008 | 14MAY2008 | 1232897220 | 1 |
| **195** | 2550 | 99999999 | 8010100009 | 10AUG2008 | 15AUG2008 | 1233484749 | 3 |
| **197** | 70201 | 99999999 | 2600100012 | 15AUG2008 | 20AUG2008 | 1233514453 | 3 |
| **206** | 49 | 121053 | 9260104510 | 10SEP2008 | 10SEP2008 | 1233689304 | 1 |

* + 1. What do you notice about the Obs colum? The Obs column keeps the observance number from the original
    2. Did the sum of **Total\_Retail\_Price** change to reflect only the subset? Yes!
  1. Add an option to suppress the Obs column.

proc print data=orion.order\_fact noobs;

sum Total\_Retail\_Price;

where Total\_Retail\_Price>500;

run;

* + 1. How can you verify the number of observations in the results? Check the log
  1. Add an ID statement to **Customer\_ID** as the identifying variable.
  2. Add a VAR statement to display **Customer\_ID, Order\_ID, Order\_Type, Quantity,** and **Total\_Retail\_Price**

proc print data=orion.order\_fact noobs;

sum Total\_Retail\_Price;

where Total\_Retail\_Price>500;

id Customer\_ID;

var Customer\_ID Order\_ID Order\_Type Quantity Total\_Retail\_Price;

run;

* + 1. What do you notice about **Customer\_ID?** Customer ID comes up twice, because it is the ID variable, and is put in in the VAR
  1. Modify the VAR statement to address the issue with **Customer\_ID**

I’ve included the last 8 rows of the result

proc print data=orion.order\_fact noobs;

sum Total\_Retail\_Price;

where Total\_Retail\_Price>500;

id Customer\_ID;

var Order\_ID Order\_Type Quantity Total\_Retail\_Price;

run;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **70187** | 1242259863 | 2 | 2 | $706.20 |
| **70165** | 1242534503 | 3 | 4 | $702.00 |
| **2806** | 1242568696 | 2 | 3 | $755.40 |
| **70100** | 1242578860 | 2 | 4 | $1,937.20 |
| **70201** | 1243165497 | 3 | 3 | $658.50 |
| **10** | 1243797399 | 1 | 2 | $1,066.40 |
| **908** | 1243887390 | 2 | 1 | $519.30 |
| **31** | 1244171290 | 1 | 3 | $760.80 |
|  |  |  |  | **$32,696.60** |

**Level 2**

1. **Displaying orion.curomer\_dim with the PRINT Procedure**

proc print data=orion.customer\_dim noobs;

where 10<=Customer\_Age<=40;

id Customer\_ID;

var Customer\_Name Customer\_Age Customer\_Type;

run;

**4.2**

**Level 1**

**5. Sorting orion.employee\_payroll and Displaying the New Data Set**

proc sort data=orion.employee\_payroll /\*sort data from orion.employee\_payroll\*/

out=sort\_salary; /\*make the new data set be called sort\_salary\*/

by Salary; /\*sort by salary\*/

proc print data=sort\_salary; /\*print the new data set\*/

run;

**6. Sorting orion.employee\_payroll and Displaying Grouped Observations**

proc sort data=orion.employee\_payroll /\*sort data from orion.employee\_payroll\*/

out=sort\_salary2; /\*make the new data set be called sort\_salary2\*/

by Employee\_Gender descending Salary; /\*sort by gender, and then descending by salary\*/

proc print data=sort\_salary2; /\*print the new data set\*/

run;

**Level 2**

**7. Sorting orion.employee\_payroll and Displaying a Subset of the New Data Set**

proc sort data=orion.employee\_payroll /\*sort the dataset orion.employee\_payroll\*/

out=sort\_sal; /\*into a new data set sort\_sal\*/

by Employee\_Gender descending Salary; /\*by gender, and then by descending salary\*/

proc print data=sort\_sal noobs; /\*print the new data set with no Obs column\*/

where Employee\_Term\_Date is null and /\*as per problem, and where salary is greater than 65000\*/

Salary>65000;

id Employee\_Gender; /\*let the ID term be gender\*/

by Employee\_Gender; /\*group by Employee\_Gender\*/

sum Salary; /\*sum over salary\*/

var Employee\_ID Salary Marital\_Status; /\*show Employee\_ID, Salary, and Marital\_Status

run;

4.3

**Level 1**

**9.**

title1 'Australian Sales Employees'; /\*set the first title as Australian Sales Employees\*/

title2 'Senior Sales Representatives'; /\*set the second title as Senior Sales Representatives\*/

footnote1 'Job\_Title: Sales Rep. IV'; /\*set a footnote\*/

proc print data=orion.sales noobs; /\*print the orion.sales data without Obs\*/

where Country='AU' and Job\_Title contains 'Rep. IV'; /\*only where Country=AU and Job Title=Rep IV\*/

var Employee\_ID First\_Name Last\_Name Gender Salary; /\*and only including these variables\*/

run;

title; /\*null title to clear all titles\*/

footnote; /\*null footnote to clear all footnotes\*/

**10.**

title 'Entry-level Sales Representatives'; /\*create title\*/

footnote 'Job\_Title: Sales Rep. I'; /\*create footnote\*/

proc print data=orion.sales noobs split='\*';/\* split='\*' is label in pt a\*, starts the print function/

where Country='US' and Job\_Title='Sales Rep. I'; /\*only prints those that satisfy these conditions\*/

var Employee\_ID First\_Name Last\_Name Gender Salary; /\*and only for these varaibles\*/

label Employee\_ID='Employee\*ID' /\*anytime you have \* here, in pt a it was just a space. This creates permanent labels for the varaibles\*/

First\_Name='First\*Name'

Last\_Name='Last\*Name'

Salary='Annual\*Salary';

run;

title; /\*null title to clear all titles\*/

footnote;/\*null footnote to clear all footnotes\*/

**Level 2**

**11.**

title1 'US Employees by State'; /\*create title\*/

proc sort data=orion.employee\_addresses /\*sort the data in orion.employee\_addresses\*/

out=ads\_by\_state; /\*and put it into this new data set\*/

where Country='US'; /\*for values where country=US only\*/

by State City Employee\_Name; /\*sort by state, then city, then employee name\*/

proc print data=ads\_by\_state noobs split='\*'; /\*print out the new data set without Obs column\*/

by State; /\*group by state\*/

where State is not null; /\*only include those values for which State is not null\*/

var Employee\_ID Employee\_Name City Postal\_Code; /\*and only for these varaibles\*/

label Employee\_Name='Name' /\*create permanent labels for these variables\*/

Postal\_Code='Zip\*Code';

run;

**Ch. 5**

**5.1**

**Level 1**

**1.**

proc print data=orion.employee\_payroll; /\*print the info from this dataset\*/

var Employee\_ID Salary Birth\_Date Employee\_Hire\_Date; /\*only for these varaibles\*/

format Salary dollar12.2 /\*format these in specific forms. Make sure to put a . after them, unless they are in decimal form!\*/

Birth\_Date MMDDYY10.

Employee\_Hire\_Date date9.;

run;

**Level 2**

**2.**

title1 'US Sales Employees'; /\*create titles\*/

title2 'Earning Under $26,000';

proc print data=orion.sales noobs split='\*'; /\*start print operation, without Obs column but with names split\*/

var Employee\_ID First\_Name Last\_Name Job\_Title Salary Hire\_Date; /\*only for these variables\*/

where Country='US' and Salary<26000; /\*only where the country is US and salary is at a threshold\*/

label First\_Name='First\*Name' /\*create permanent labels for these varaibles. Use \* to indicated that we want it on two lines instead of just one\*/

Last\_Name='Last Name'

Job\_Title='Title'

Hire\_Date='Date Hired';

format Salary dollar12. /\*format the salary and date variables specifically\*/

Hire\_Date MONYY7.

run;

title; /\*null title to clear all titles\*/

**5.2**

**Level 1**

proc format; /\*format function for Gender as female or male\*/

value $GENDER 'F'='Female'

'M'='Male';

proc format; /\*format function for month\*/

value MNAME 1='January'

2='February'

3='March';

data Q1Birthdays; /\*create data set Q1Birthdays\*/

set orion.employee\_payroll; /\*put orion.employee\_payroll into it\*/

BirthMonth=month(Birth\_Date); /\*create the variable BirthMonth by using the month function, which returns a number\*/

if BirthMonth le 3; /\*if BirthMonth<=3, so beforemarch\*/

title ‘Employees with Birthdays in Q1’; /\*create title\*/

proc print data=work.Q1Birthdays; /\*print the new dataset\*/

format Employee\_Gender $GENDER. /\*use the format functions to print in the way we want\*/

BirthMonth MNAME.;

run;

**Level 2**

**5.**

proc format; /\*format functions for gender and salary range\*/

value $GENDER 'F'='Female'

'M'='Male'

other='Invalid code';

proc format;

value SALRANGE 20000-<100000='Below $100,000'

100000-<500000='$100,000 or more'

.='Missing salary' /\*a null value is indicated by .\*/

other='Invalid salary'; /\*can write other for any other value\*/

proc print data=orion.nonsales; /\*start print\*/

var Employee\_ID Job\_Title Salary Gender;/\*only for these varaibles\*/

title1 'Salary and Gender Values'; /\*can create titles inside the print function too\*/

title2 'for Non-Sales Employees';

format Gender$GENDER. /\*apply our gender and salary format functions to those variables to make it look like we want\*/

Salary SALRANGE.;

run;

**Ch 6.2**

**Level 2**

data work.delays; /\*create the data set work.delays\*/

set orion.orders; /\*import and use the data from orion.orders\*/

where 4<Delivery\_Date-Order\_Date /\*in the question, we only wanted to keep data where the Delviary Date is more than four days past the Order\_Date\*/

and Employee\_ID=99999999; /\*and where Employee\_ID=99999999\*/

Order\_Month=month(Order\_Date); /\*Create the new variable Order\_Month, using the month function to extract the info from Order\_Date\*/

if Order\_Month=8;/\*Only want to keep August, and since the month function returns a number 1-12 corresponding to the months, we want only the 8th month\*/

keep Employee\_ID Customer\_ID Order\_Date Delivery\_Date Order\_Month; /\*we want to keep only these parameters, all others will drop out\*/

label Order\_Date='Date Ordered' /\*Create new labels for the variables\*/

Delivery\_Date='Date Delivered'

Order\_Month='Month Ordered';

format Order\_Date MMDDYY10. /\*want to format these variables in this form\*/

Delivery\_Date MMDDYY10.;

run;

proc contents data=work.delays; /\*run a proc contents as per the question, of the new data set\*/

run;

proc print data=work.delays; /\* don't include label, which makes use of our permanent labels, becasue that is not what we are asked for\*/

run;

**Ch 9**

**9.1**

**Level 2**

data work.birthday; /\*create new data set work.birthday\*/

set orion.customer; /\*use the data from orion.customer\*/

Bday2012=mdy(month(Birth\_Date), day(Birth\_Date), 2102); /\*create new variable with the cool mdy function, which takes the month of birth\_date, the day of birth\_date, and the common year 2012 and makes it a date\*/

BdayDOW2012=weekday(Bday2012); /\*creates new variable with the weekday function applied to Bday2012, to get the day of the week of your birthday in 2012\*/

Age2012=(Bday2012-Birth\_Date)/365.25;/\*creates new variable Age2012 by subtracting your birthdate from your birthday 2012, and dividing by 365.25\*/

keep Customer\_Name Birth\_Date Bday2102 BdayDOW2012 Age2012; /\*keep only these variables\*/

format Bday2012 Date9. /\*use this format for these dates and ages\*/

Age2012 3.;

proc print data=work.birthday; /\*print the new data set work.birthday. THIS ACTUALLY DOESN’T WORK, because there is a text formatting issue with orion.customer, which was formatted for DOS. So in our VM, the format turns to mush that the program can’t deal with.\*/

run;

**9.2**

**6.**

data work.season; /\*create new dataset work.season\*/

set orion.customer\_dim; /\*use the data from orion.customer\_dim\*/

if qtr(Customer\_BirthDate)=1 then \*use the if-else-then method to check if BirthDate is in the ast, 2nd, 3rd or 4th quarter. Use the nifty qtr function, which assigns 1-4 to a BirthDate, to do this. Then, create the variable Promo\*/

Promo='Winter';

else if qtr(Customer\_BirthDate)=2 then

Promo='Spring';

else if qtr(Customer\_BirthDate)=3 then

Promo='Summer';

else if qtr(Customer\_BirthDate)=4 then

Promo='Fall';

if 18=<Customer\_Age<=25 then /\*another if-else-then which tests age and creates a variable Promo2 based on the results\*/

Promo2='YA';

else if Customer\_Age>=65 then

Promo2='Senior';

keep Customer\_FirstName Customer\_LastName Customer\_BirthDate Customer\_Age Promo Promo2; /\*keep only these variables in the data set\*/

run;

proc print data=work.season; /\*print the new data set work.birthday. THIS ACTUALLY DOESN’T WORK, because there is a text formatting issue with orion.customer, which was formatted for DOS. So in our VM, the format turns to mush that the program can’t deal with.\*/

run;

**7.**

data work.ordertype; /\*create new data set\*/

set orion.orders; /\*set it up with all orion.orders data\*/

DayOfWeek=weekday(Order\_Date); /\*create new variable with very cool function weekday, which assigns a number of each day of the week\*/

if Order\_Type=1 then /\*use the if-else-then-do formatting to create new variables Retail Sale and SaleAds, which take on different values based on the value of Order\_Type. For the first if, we do not need a do command because we are only issuing one command. Because we want to create two variables for each of the other values of Order\_Type, however, we need to issue a do command, and put an end; at the finish\*/

Type='Retail Sale';

else if Order\_Type=2 then do;

Type='Catalog Sale';

SaleAds='Mail';

end;

else if Order\_Type=3 then do;

Type='Internet Sale';

SaleAds='Email';

end;

drop Order\_Type Employee\_ID Customer\_ID; /\*include all varaibles BUT these. The opposite of a keep\*/

run;

proc print data=work.ordertype; /\*print the new dataset\*/

run;

**Ch. 10**

**10.1**

**Level 2**

**3.**

**a)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Code |  | Company |  | ContactType |  |
|  | Type | Length | Type | Length | Type | Length |
| Orion.charities | Char | 6 | Char | 40 | char | 10 |
| Orion.us\_suppliers | Char | 6 | Char | 30 | Char | 1 |
| Orion.consultants | Char | 6 | Char | 30 | Num | 8 |

b/c)

data contacts;

set orion.charities orion.us\_suppliers;

run;

proc contents data=contacts;

run;

| **Alphabetic List of Variables and Attributes** | | | |
| --- | --- | --- | --- |
| **#** | **Variable** | **Type** | **Len** |
| **3** | Code | Char | 6 |
| **1** | Company | Char | 40 |
| **2** | ContactType | Char | 10 |
|  |  |  |  |

The input data was assigned from orion.charities, the first dataset I wrote.

d/e)

data contacts2;

set orion.us\_suppliers orion.charities;

run;

proc contents data=contacts2;

run;

| **Alphabetic List of Variables and Attributes** | | | |
| --- | --- | --- | --- |
| **#** | **Variable** | **Type** | **Len** |
| **1** | Code | Char | 6 |
| **2** | Company | Char | 30 |
| **3** | ContactType | Char | 1 |

The input data was assigned from orion.us\_suppliers, the first dataset I wrote.

f/g)

data contacts3;

set orion.us\_suppliers orion.consultants;

run;

proc contents data=contacts3;

This failed because the variable ContactType is characterized as a character for orion.us\_suppliers and as numeric for orion.consultants.

**10.3**

**Level 2**

**5.**

proc sort data-=orion.product\_list /\*sort orion.product\_list\*/

out=work.product\_list; /\*creating the new dataset work.product\_list\*/

by Product\_Level; /\*sort by Product\_level\*/

run;

\*proc print data=work.product\_list;

\*run;

data work.listlevel; /\*create new data set\*/

merge orion.product\_level work.product\_list; /\*make it the merge between these two sets\*/

keep Product\_ID Product\_Name Product\_Level Product\_Level\_Name; /\*only for these variables\*/

run;

proc print data=work.listlevel noobs; /\*print the dataset without Obs column\*/

where Product\_Level=3; /\*only for this condition\*/

run;

**10.4**

**Level 2**

**8.**

proc sort data=orion.customer /\*sort the dataset orion.customer\*/

out=work.customer; /\*and make a new dataset work.customer to hold this info\*/

by Country; /\*by country\*/

run;

data work.allcustomer; /\*create new dataset work.allcustomer\*/

merge work.customer (in=Cust) /\*by merging all data in work.customer and orion.lookup\_country. Cust becomes 1 if the dataset is used to create the specific observation\*/

orion.lookup\_country(rename=(Start=Country Label=Country\_Name)/\*and renames the variable Country as Country\_Name

by Country; /\*group by country\*/

keep Customer\_ID Country Customer\_Name Country\_Name; /\*keep only these variables\*/

if Cust=1 and CTRY=1; /\*only if Cust(from above)=1 and CTRY=1,\*/

run;

proc print data=work.allcustomer; /\*print the new dataset\*/

run;