Chapter 4: Producing Detail Reports

Chapter 4: Producing Detail Reports

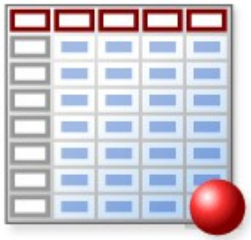
Objectives

- Create a default PROC PRINT report.
- Select variables with a VAR statement.
- Calculate totals with a SUM statement.
- Select observations with a WHERE statement.
- Define a date constant.
- Identify observations with an ID statement.

Business Scenario

Orion Star management wants a report that displays the names, salaries, and a salary total for all sales employees.

orion.sales



PROC PRINT

Obs	Last_Name	First_Name	Salary
1	xxxxxxx	xxxxxxxxxx	99999
2	xxxxxxx	xxxxxxxxxx	99999
3	xxxxxxx	xxxxxxxxxx	99999

			99999

PRINT Procedure

By default, PROC PRINT displays all observations, all variables, and an Obs column on the left side.

```
proc print data=orion.sales;
run;
```

Partial PROC PRINT Output

Obs	Employee_ID	First_ Name	Last_Name	Gender	Salary	Job_Title	Country	Birth_ Date	Hire_ Date
1	120102	Tom	Zhou	M	108255	Sales Manager	AU	3510	10744
2	120103	Wilson	Dawes	M	87975	Sales Manager	AU	-3996	5114
3	120121	Irenie	Elvish	F	26600	Sales Rep. II	AU	-5630	5114
4	120122	Christina	Ngan	F	27475	Sales Rep. II	AU	-1984	6756
5	120123	Kimiko	Hotstone	F	26190	Sales Rep. I	AU	1732	9405

Statements and options can be added to the PRINT procedure to modify the default behavior.

p104d01

VAR Statement

The VAR statement selects variables to include in the report and specifies their order.

```
proc print data=orion.sales;
   var Last_Name First_Name Salary;
run;
```

VAR *variable(s)*;

Partial PROC PRINT Output

Obs	Last_Name	First_ Name	Salary
1	Zhou	Tom	108255
2	Dawes	Wilson	87975
3	Elvish	Irenie	26600
4	Ngan	Christina	27475
5	Hotstone	Kimiko	26190

p104d01

SUM Statement

The *SUM statement* calculates and displays report totals for the requested ***numeric*** variables.

```
proc print data=orion.sales;
   var Last_Name First_Name Salary;
   sum Salary;
run;
```

SUM *variable(s)*;

Partial PROC PRINT Output

Obs	Last_Name	First_ Name	Salary
1	Zhou	Tom	108255
2	Dawes	Wilson	87975
3	Elvish	Irenie	26600
...			
164	Capachietti	Renee	83505
165	Lansberry	Dennis	84260
		=======	
		5141420	

p104d01

Viewing the Log

Partial SAS Log

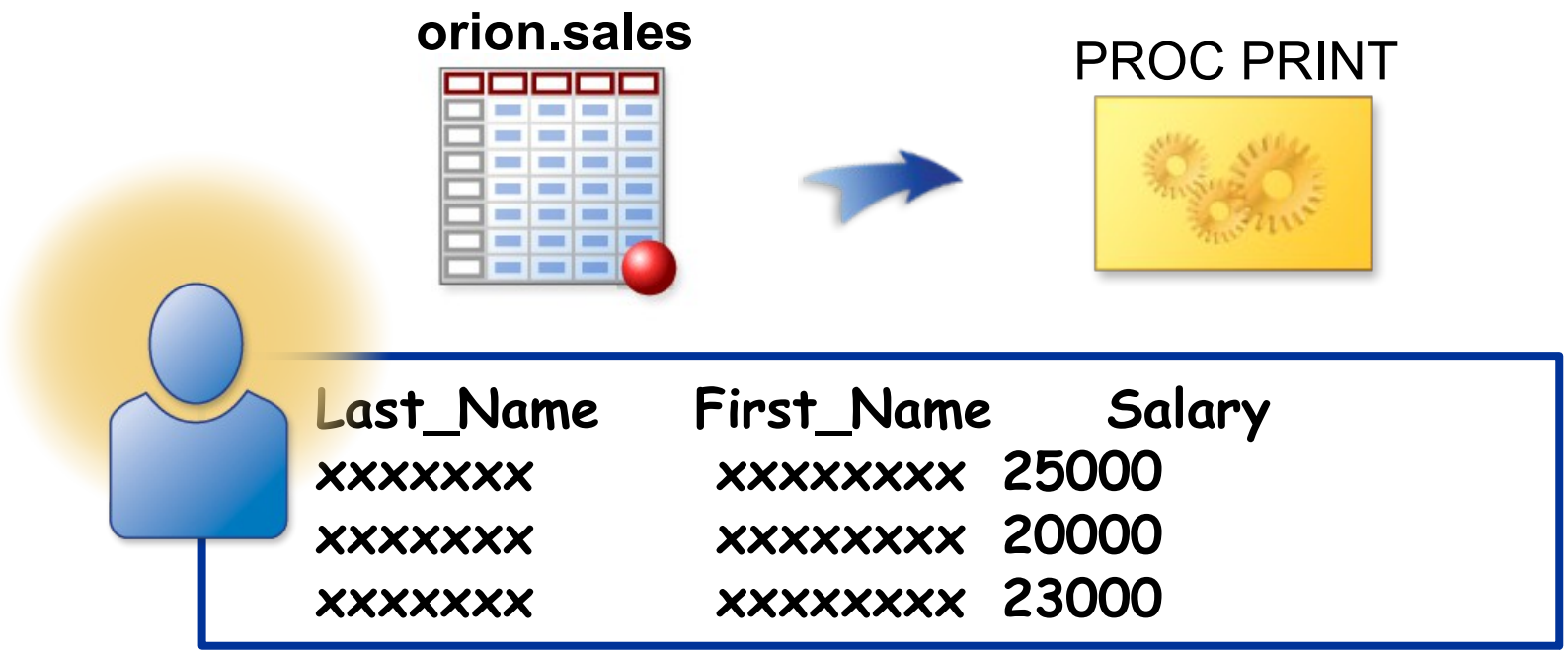
```
84   proc print data=orion.sales;
85      var Last_Name First_Name Salary;
86      sum salary;
87   run;

NOTE: There were 165 observations read from the data set ORION.SALES.
```

- The order of statements in a SAS procedure is usually not important.

Business Scenario

Orion Star management wants a report that displays the names and salaries of the sales employees earning less than $25,500. Suppress the Obs column.



WHERE Statement

The *WHERE statement* selects observations that meet the criteria specified in the WHERE expression.

```
proc print data=orion.sales;
   var Last_Name First_Name Salary;
   where Salary<25500;
run;
```

WHERE *WHERE-expression*;

p104d02

Viewing the Log

Only 7 of the 165 observations from **orion.sales** were selected by the WHERE statement.

```
295  proc print data=orion.sales;
296     var Last_Name First_Name Salary;
297     where Salary<25500;
298  run;

NOTE: There were 7 observations read from the data set ORION.SALES.
      WHERE Salary<25500;
```

Viewing the Output

PROC PRINT Output

Obs	Last_ Name	First_ Name	Salary
49	Tilley	Kimiko	25185
50	Barcoe	Selina	25275
85	Anstey	David	25285
104	Voron	Tachaun	25125
111	Polky	Asishana	25110
131	Ould	Tulsidas	22710
148	Buckner	Burnetta	25390

original observation numbers

Suppressing the Obs Column

Use the NOOBS option in the PROC PRINT statement to suppress the Obs column.

```
proc print data=orion.sales noobs;
   var Last_Name First_Name Salary;
   where Salary<25500;
run;
```

PROC PRINT DATA=*SAS-data-set* **NOOBS;**

PROC PRINT Output

Last_ Name	First_ Name	Salary
Tilley	Kimiko	25185
Barcoe	Selina	25275
Anstey	David	25285
Voron	Tachaun	25125
Polky	Asishana	25110
Ould	Tulsidas	22710
Buckner	Burnetta	25390

p104d02

WHERE Statement

The WHERE expression defines the condition (or conditions) for selecting observations.

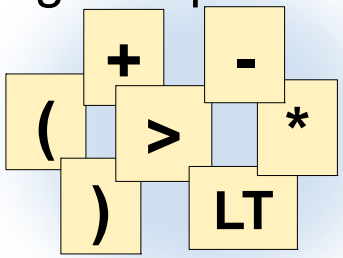
WHERE *WHERE-expression*;

Operands

- character constants
- numeric constants
- date constants
- character variables
- numeric variables

Operators

- symbols that represent a comparison, calculation, or logical operation



- SAS functions
- special WHERE operators

Operands

Constants are fixed values.

- Character values are enclosed in quotation marks and are case sensitive.
- Numeric values do not use quotation marks or special characters.

Variables must exist in the input data set.

```
where Gender='M';
```

variable constant

```
where Salary>50000;
```

variable constant

SAS Date Constant

A *SAS date constant* is a date written in the following form: **'ddmmm<yy>yy'd**

Examples
'01JAN2000'd
'31Dec11'D
'1jan04'd
'06NOV2000'D

SAS automatically converts a date constant to a SAS date value.

Comparison Operators

Comparison operators compare a variable with a value or with another variable.

Symbol	Mnemonic	Definition
=	EQ	Equal to
^= ¬= ~=	NE	Not equal to
>	GT	Greater than
<	LT	Less than
>=	GE	Greater than or equal
<=	LE	Less than or equal
	IN	Equal to one of a list

Comparison Operators

Examples
`where Gender eq ' ';`
`where Salary ne .;`
`where Salary>=50000;`
`where Hire_Date<'01Jan2000'd;`
`where Country in ('AU','US');`
`where Country in ('AU' 'US');`
`where Order_Type in (1,2,3);`

The value list in the IN operator must be enclosed in parentheses and separated by either commas or blanks. Character values must be enclosed in quotation marks.

Setup for the Poll

Program **p104a01** contains two WHERE statements. Open and submit the program.

4.01 Multiple Choice Poll

Which of the following is true?

- The program executes, applying both WHERE conditions successfully.
- The program fails and an error message is written to the log.
- The program executes, but only the first WHERE condition is applied.
- The program executes, but only the second WHERE condition is applied.

4.01 Multiple Choice Poll – Correct Answer

Which of the following is true?

- The program executes, applying both WHERE conditions successfully.
- The program fails and an error message is written to the log.
- The program executes, but only the first WHERE condition is applied.
- The program executes, but only the second WHERE condition is applied.

```
182  proc print data=orion.sales;
183     where Country='AU';
184     where Salary<30000;
NOTE: WHERE clause has been replaced.
185  run;

NOTE: There were 134 observations read from the data set ORION.SALES.
      WHERE Salary<30000;
```

Logical Operators

Logical operators combine or modify WHERE expressions.

```
proc print data=orion.sales;
   where Country='AU' and
         Salary<30000;
run;
```

WHERE *WHERE-expression-1* AND | OR *WHERE-expression-n*;

p104d03

Viewing the Log

Partial SAS Log

```
67   proc print data=orion.sales;
68      where Country='AU' and
69            Salary<30000;
70   run;

NOTE: There were 51 observations read from the data set ORION.SALES.
WHERE (Country='AU') and (Salary<30000);
```

Logical Operator Priority

The operators can be written as symbols or mnemonics, and parentheses can be added to modify the order of evaluation.

Symbol	Mnemonic	Priority
^ ¬ ~	NOT	I
&	AND	II
|	OR	III

The NOT operator modifies a condition by finding the complement of the specified criteria.

```
where City not in ('London','Rome','Paris');
```

Logical Operators

Examples
`where Country ne 'AU' and Salary>=50000;`
`where Gender eq 'M' or Salary ge 50000;`
`where Country='AU' or Country='US';`
`where Country in ('AU','US');`
`where Country not in ('AU','US');`

equivalent expressions

You should use only one WHERE statement in a step.

4.02 Quiz

Which WHERE statement correctly subsets the numeric values for May, June, or July and missing character names?

```
where Month in (5-7)
      and Names=.;
```

```
where Month in (5,6,7)
      and Names=' ';
```

```
where Month in ('5','6','7')
      and Names='.';
```

4.02 Quiz – Correct Answer

Which WHERE statement correctly subsets the numeric values for May, June, or July and missing character names?

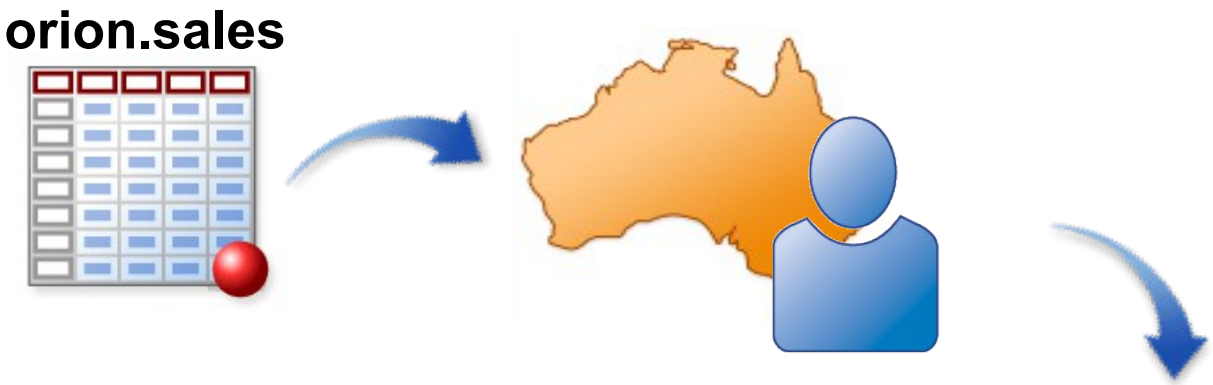
```
where Month in (5-7)
      and Names=.;
```

```
where Month in (5,6,7)
      and Names=' ';
```

```
where Month in ('5','6','7')
      and Names='.';
```

Business Scenario

Orion Star management wants a report that lists only the Australian sales representatives.



Last_Name	First_ Name	Country	Job_Title
xxxxxxxxxx	xxxxxxx	xx	xxxxxxxxxxxx
xxxxxxxxxx	xxxxxxx	xx	xxxxxxxxxxxx
xxxxxxxxxx	xxxxxxx	xx	xxxxxxxxxxxx
xxxxxxxxxx	xxxxxxx	xx	xxxxxxxxxxxx

Exploring the Data

```
proc print data=orion.sales noobs;
   var Last_Name First_Name Country
       Job_Title;
run;
```

Partial PROC PRINT Output

Plested	Billy	AU	Sales Rep. II
Wills	Matsuoka	AU	Sales Rep. III
George	Vino	AU	Sales Rep. II
Body	Meera	AU	Sales Rep. III
Highpoint	Harry	US	Chief Sales Officer
Magolan	Julienne	US	Sales Rep. II
Desanctis	Scott	US	Sales Rep. IV
Ridley	Cherda	US	Sales Rep. IV

p104d04

Subsetting in a PROC PRINT Step

Include a WHERE statement to subset by **Country** and **Job_Title**.

```
proc print data=orion.sales noobs;
   var Last_Name First_Name Country
       Job_Title;
   where Country='AU' and
         Job_Title contains 'Rep';
run;
```

CONTAINS is a special WHERE operator.

p104d04

CONTAINS Operator

The *CONTAINS operator* selects observations that include the specified substring.

Equivalent Statements
`where Job_Title contains 'Rep';`
`where Job_Title ? 'Rep';`

- ? can be used instead of the mnemonic.
- The position of the substring within the variable's values is not important.
- Comparisons made with the CONTAINS operator are case sensitive.

Viewing the Output

Partial PROC PRINT Output

Last_Name	First_ Name	Country	Job_Title
Elvish	Irenie	AU	Sales Rep. II
Ngan	Christina	AU	Sales Rep. II
Hotstone	Kimiko	AU	Sales Rep. I
Daymond	Lucian	AU	Sales Rep. I
Hofmeister	Fong	AU	Sales Rep. IV

Special WHERE Operators

Special WHERE operators are operators that can be used only in WHERE expressions.

Operator	Definition	Char	Num
CONTAINS	Includes a substring	x	
BETWEEN-AND	An inclusive range	x	x
WHERE SAME AND	Augment a WHERE expression	x	x
IS NULL	A missing value	x	x
IS MISSING	A missing value	x	x
LIKE	Matches a pattern	x	

BETWEEN-AND Operator

The *BETWEEN-AND operator* selects observations in which the value of a variable falls within an inclusive range of values.

Examples
`where salary between 50000 and 100000;`
`where salary not between 50000 and 100000;`
`where Last_Name between 'A' and 'L';`
`where Last_Name between 'Baker' and 'Gomez';`

BETWEEN-AND Operator

Equivalent Statements
`where salary between 50000 and 100000;`
`where salary>=50000 and salary<=100000;`
`where 50000<=salary<=100000;`

WHERE SAME AND Operator

Use the *WHERE SAME AND operator* to add more conditions to an existing WHERE expression.

```
proc print data=orion.sales;
   where Country='AU' and Salary <30000;
   where same and Gender='F';
   var First_Name Last_Name Gender
       Salary Country;
run;
```

The WHERE SAME AND condition ***augments*** the original condition.

p104d05

Viewing the Log

Partial SAS Log

```
22   proc print data=orion.sales;
23       where Country='AU'  and Salary<30000;
24       where also Gender='F';
NOTE: WHERE clause has been augmented.
25       var First_Name Last_Name Gender Salary Country;
26   run;

NOTE: There were 23 observations read from the data set ORION.SALES.
     WHERE (Country='AU') and (Gender='F') and (Salary<30000);
```

Viewing the Output

Partial PROC PRINT Output

Obs	First_Name	Last_Name	Gender	Salary	Country
3	Irenie	Elvish	F	26600	AU
4	Christina	Ngan	F	27475	AU
5	Kimiko	Hotstone	F	26190	AU
9	Sharryn	Clarkson	F	28100	AU
14	Fancine	Kaiser	F	28525	AU
15	Petrea	Soltau	F	27440	AU
19	Marina	Iyengar	F	29715	AU
20	Shani	Duckett	F	25795	AU
21	Fang	Wilson	F	26810	AU
23	Amanda	Liebman	F	27465	AU

Country='AU'
Gender='F'
Salary<30000

4.03 Quiz

- Open **p104a01b**.
- Change WHERE SAME AND to WHERE ALSO.
- Submit the program and view the log.

What message is written to the log?

4.03 Quiz – Correct Answer

WHERE ALSO results in the same message:

`WHERE clause has been augmented.`

```
27   proc print data=orion.sales;
28      where Country='AU'  and Salary<30000;
29      where also Gender='F';
NOTE: WHERE clause has been augmented.
30      var First_Name Last_Name Gender Salary Country;
31   run;

NOTE: There were 23 observations read from the data set ORION.SALES.
    WHERE (Country='AU') and (Gender='F') and (Salary<30000);
```

IS NULL Operator

The *IS NULL operator* selects observations in which a variable has a missing value.

Examples
`where Employee_ID is null;`
`where Employee_ID is not null;`

IS NULL can be used for both character and numeric variables, and is equivalent to the following statements:

```
where employee_ID=' ';
```

```
where employee_ID=.;
```

IS MISSING Operator

The *IS MISSING operator* selects observations in which a variable has a missing value.

Examples
`where Employee_ID is missing;`
`where Employee_ID is not missing;`

IS MISSING can be used for both character and numeric variables, and is equivalent to the following statements:

```
where employee_ID=' ';
```

```
where employee_ID=.;
```

LIKE Operator

The *LIKE operator* selects observations by comparing character values to specified patterns. Two special characters are used to define a pattern:

- A percent sign (%) specifies that ***any number*** of characters can occupy that position.
- An underscore (_) specifies that ***exactly one*** character can occupy that position.

Examples
`where Name like '%N';`
`where Name like 'T_m';`
`where Name like 'T_m%';`

4.04 Quiz

Which WHERE statement returns all the observations that have a first name starting with the letter M for the given values?

```
where Name like '_, M_';
```

```
where Name like '%, M%';
```

```
where Name like '_, M%';
```

– Answer

```
where Name like '%, M_';
```

– Answer

Name
Elvish, Irenie
Ngan, Christina
Hotstone, Kimiko
Daymond, Lucian
Hofmeister, Fong
Denny, Satyakam
Clarkson, Sharryn
Kletschkus, Monica

last name, first name

4.04 Quiz – Correct Answer

Which WHERE statement returns all the observations that have a first name starting with the letter M for the given values?

```
where Name like '_, M_';
```

```
where Name like '%, M%';
```

```
where Name like '_, M%';
```

– Answer

```
where Name like '%, M_';
```

– Answer

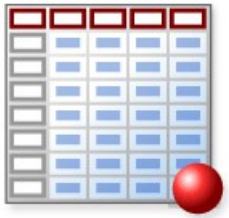
Name
Elvish, Irenie
Ngan, Christina
Hotstone, Kimiko
Daymond, Lucian
Hofmeister, Fong
Denny, Satyakam
Clarkson, Sharryn
Kletschkus, Monica

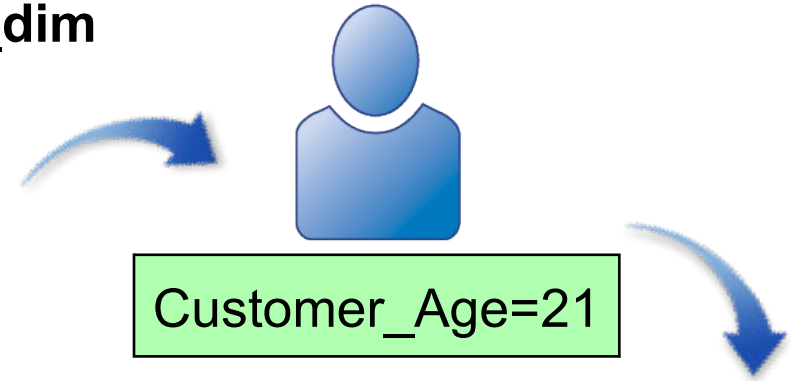
last name, first name

Business Scenario

The Sales Manager wants a report that includes only customers who are 21 years old.

orion.customer_dim





Obs	Customer_ID	Customer_Name	Customer_ Gender	Customer_ Country	Customer_Group	Customer_ Age_Group	Customer_ Type
1	999	XXXXXXXXXXXX	X	XX	XXXXXXXXXXXX	XXXXXXXXX	XXXXXXXX
2	999	XXXXXXXXXXXX	X	XX	XXXXXXXXXXXX	XXXXXXXXX	XXXXXXXX
3	999	XXXXXXXXXXXX	X	XX	XXXXXXXXXXXX	XXXXXXXXX	XXXXXXXX

Subsetting the Data Set

Display the required rows and variables.

```
proc print data=orion.customer_dim;
   where Customer_Age=21;
   var Customer_ID Customer_Name
       Customer_Gender Customer_Country
       Customer_Group Customer_Age_Group
       Customer_Type;
run;
```

- The subsetting variable does not need to be included in the report.

p104d06

Viewing the Output

In this output, two lines are used for each observation.

PROC PRINT Output

Obs	Customer_ID	Customer_ Customer_Name	Gender	Country	Customer_Group
37	79	Najma Hicks	F	US	Orion Club members
58	11171	Bill Cuddy	M	CA	Orion Club Gold members
66	46966	Lauren Krasowski	F	CA	Orion Club members
...					
76	70210	Alex Santinello	M	CA	Orion Club members

Obs	Customer_ Age_Group	Customer_Type
37	15-30 years	Orion Club members medium activity
58	15-30 years	Orion Club Gold members low activity
66	15-30 years	Orion Club members high activity
...		
76	15-30 years	Orion Club members medium activity

The Obs column helps identify observations in a report that span multiple lines.

ID Statement

The *ID statement* specifies the variable or variables to print at the beginning of each row instead of an observation number.

```
proc print data=orion.customer_dim;
   where Customer_Age=21;
   id Customer_ID;
   var Customer_Name Customer_Gender
       Customer_Country Customer_Group
       Customer_Age_Group Customer_Type;

run;
```

Choose ID variables that uniquely identify observations.

p104d07

Viewing the Output

PROC PRINT Output

Customer_ID	Customer_Name	Gender	Customer_Country	Customer_Group
79	Najma Hicks	F	US	Orion Club members
11171	Bill Cuddy	M	CA	Orion Club Gold members
46966	Lauren Krasowski	F	CA	Orion Club members
70079	Lera Knott	F	CA	Orion Club members
70187	Soberina Berent	F	CA	Orion Club members
70210	Alex Santinello	M	CA	Orion Club members

Customer_ID	Customer_Age_Group	Customer_Type
79	15-30 years	Orion Club members medium activity
11171	15-30 years	Orion Club Gold members low activity
46966	15-30 years	Orion Club members high activity
70079	15-30 years	Orion Club members medium activity
70187	15-30 years	Orion Club members medium activity
70210	15-30 years	Orion Club members medium activity

Exercise

This exercise reinforces the concepts discussed previously.

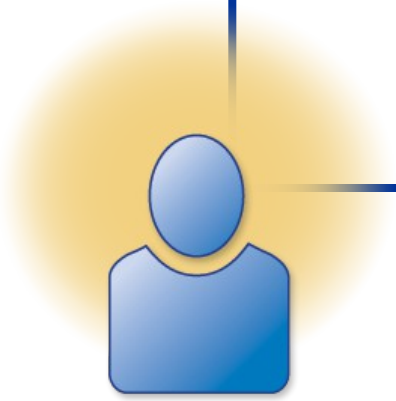
Chapter 4: Producing Detail Reports

Objectives

- Sort the observations in a SAS data set based on the values of one or more variables.
- Display the sorted observations.
- Display a data set with report totals and subtotals for each BY group.

Business Scenario

Display observations from **orion.sales** in ascending order by the variable **Salary**.



Employee_ID	Last_Name	Salary
999999	xxxxxxxxxx	99999
999999	xxxxxxxxxx	99999
999999	xxxxxxxxxx	99999

Creating a Sorted Report

Step 1	Use the SORT procedure to create a new data set, **work.sales**, ordering the observations by the value of **Salary**.

Creating a Sorted Report

Step 2	Use the PRINT procedure to display the sorted data set, **work.sales**.

Step 1: SORT Procedure

The *SORT procedure* rearranges the observations in the input data set based on the values of the variable or variables listed in the BY statement.

```
proc sort data=orion.sales
          out=work.sales;
   by Salary;
run;
```

PROC SORT DATA=*input-SAS-data-set*
<**OUT=***output-SAS-data-set*>;
BY <**DESCENDING**> *variable*<*s*>;
RUN;

The BY statement in a PROC SORT step specifies the sort variables and, optionally, the sort order.

p104d08

Viewing the Log

The SORT procedure does not produce a report. Check the log for errors or warnings.

Partial SAS Log

```
34   proc sort data=orion.sales
35           out=work.sales;
36     by Salary;
37   run;

NOTE: There were 165 observations read from the data set ORION.SALES.
NOTE: The data set WORK.SALES has 165 observations and 9 variables.
```

Step 2: Viewing the Output

```
proc print data=work.sales noobs;
   var Employee_ID Last_Name Salary;
run;
```

Partial PROC PRINT Output

Employee_ID	Last_Name	Salary
121084	Ould	22710
121064	Polky	25110
121057	Voron	25125
...		
121143	Favaron	95090
120102	Zhou	108255
120261	Highpoint	243190

p104d08

SORT Procedure

The SORT procedure

- replaces the original data set or creates a new one
- can sort on multiple variables
- sorts in ascending (default) or descending order
- does not generate printed output.

The input data set is overwritten unless the OUT= option is used to specify an output data set.

4.05 Quiz

Which step sorts the observations in a SAS data set and overwrites the same data set?

```
proc sort data=work.EmpsAU
          out=work.sorted;
   by First;
run;
```

```
proc sort data=orion.EmpsAU
          out=EmpsAU;
   by First;
run;
```

```
proc sort data=work.EmpsAU;
   by First;
run;
```

4.05 Quiz – Correct Answer

Which step sorts the observations in a SAS data set and overwrites the same data set?

```
proc sort data=work.EmpsAU
          out=work.sorted;
   by First;
run;
```

```
proc sort data=orion.EmpsAU
          out=EmpsAU;
   by First;
run;
```

```
proc sort data=work.EmpsAU;
   by First;
run;
```

Business Scenario

Produce a report that lists sales employees grouped by **Country**, in descending **Salary** order within country.

```
-------------------------------------Country=AU-----------------------------------------

                 First_   Last_                                    Birth_   Hire_
  Employee_ID    Name     Name     Gender   Salary    Job_Title    Date     Date

     9999        xxxx     xxxxx     x       99999     xxxxxx       9999     9999
     9999        xxxx     xxxxx     x       99999     xxxxxx       9999     9999


-------------------------------------Country=US-----------------------------------------

                 First_   Last_                                    Birth_   Hire_
  Employee_ID    Name     Name     Gender   Salary    Job_Title    Date     Date

     9999        xxxx     xxxxx     x       99999     xxxxxx       9999     9999
     9999        xxxx     xxxxx     x       99999     xxxxxx       9999     9999
     9999        xxxx     xxxxx     x       99999     xxxxxx       9999     9999
```

Creating a Grouped Report

Step 1	Use the SORT procedure to group data in a data set. This scenario requires two variables to be sorted: • **Country** • descending **Salary** within **Country**
Step 2	Use a BY statement in PROC PRINT to display the sorted observations grouped by **Country**.

Step 1: Sort the Data

Sort the data set to group the observations.

```
proc sort data=orion.sales
        out=work.sales;
   by Country descending Salary;
run;
```

BY <**DESCENDING**> *variable(s)*;

p104d09

Specifying Sort Order

The *DESCENDING option* reverses the sort order for the variable that immediately follows it. The observations are sorted from the largest value to the smallest value.

Examples:

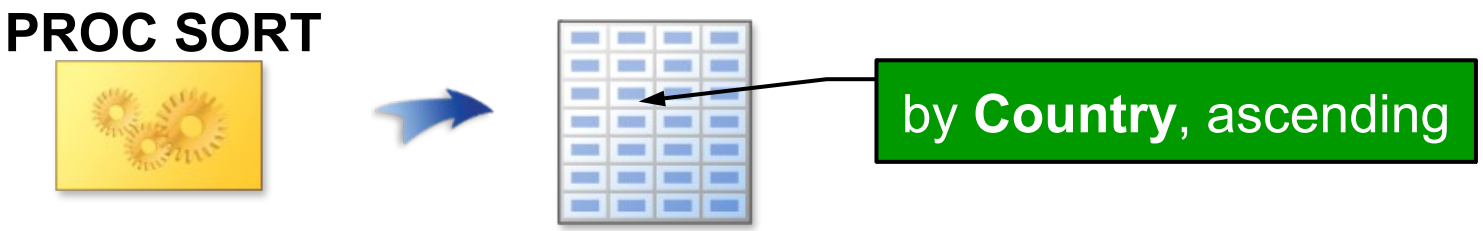
```
by descending Last First;
```

```
by Last descending First;
```

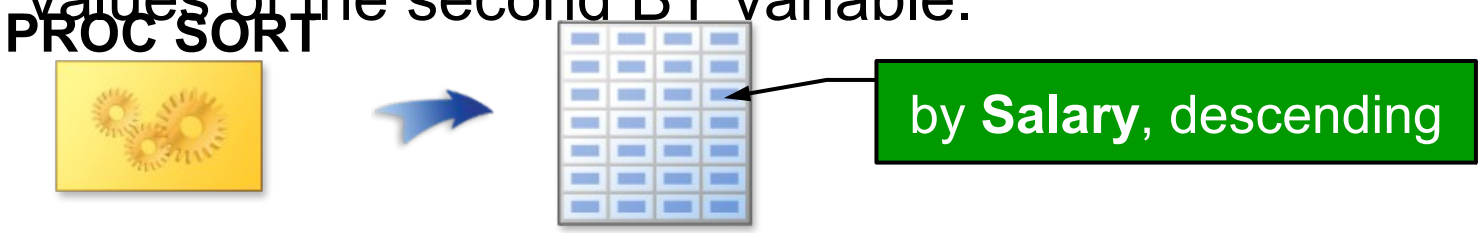
```
by descending Last descending First;
```

Specifying Multiple BY Variables

- PROC SORT first arranges the data set by the values of the first BY variable.



- PROC SORT then arranges any observations that have the same value of the first BY variable by the values of the second BY variable.



- This sorting continues for every specified BY variable.

Step 2: Specify Report Groupings

The BY statement in a PROC PRINT step specifies the variable or variables to use to form *BY groups*.

```
proc print data=work.sales noobs;
   by Country;
run;
```

BY <**DESCENDING**> *variable(s)*>**;**

- The variables in the BY statement are called *BY variables*.
- The observations in the data set ***must*** be in order by the BY variable (or variables).

p104d09

Viewing the Output

Partial PROC PRINT Output

```
---------------------------- Country=AU --------------------------

                                                Hire_
Employee_ID   First_Name   Last_Name  Gender  Salary        Date

    120102   Tom         Zhou         M    108255        12205
    120103   Wilson       Dawes        M     87975  ...    6575
    120168   Selina       Barcoe       M     36605        18567

---------------------------- Country=US --------------------------

                                                Hire_
Employee_ID   First_Name   Last_Name   Gender  Salary        Date

    120261   Harry       Highpoint     M     243190        11535
    121143   Louis       Favaron       M      95090   ...  15157
    121064   Asishana     Polky        M      84260        13027
```

4.06 Quiz

Open and submit **p104a02**. View the log.

Why did the program fail?

4.06 Quiz – Correct Answer

Open and submit **p104a02**. View the log.

Why did the program fail?

The input data set was not sorted by Gender.

```
188  proc sort data=orion.sales
189          out=work.sorted;
190     by Country Gender;
191  run;

NOTE: There were 165 observations read from the data set ORION.SALES.
NOTE: The data set WORK.SORTED has 165 observations and 9 variables.

192
193  proc print data=work.sorted;
194     by Gender;
195  run;

ERROR: Data set WORK.SORTED is not sorted in ascending sequence. The current
       BY group has Gender = M and the next BY group has Gender = F.
NOTE: The SAS System stopped processing this step because of errors.
NOTE: There were 64 observations read from the data set WORK.SORTED.
```

Business Scenario

Modify the previous report to display selected variables, the salary subtotal for each country, and the salary grand total.

```
-------------------- Country=AU ------------------

First_Name   Last_Name     Gender    Salary

XXXX          XXXXXXX       X        99999
XXXX          XXXXXXX       X        99999
--------                             ---------
Country                              999999

-------------------- Country=US ------------------

First_Name   Last_Name     Gender    Salary

XXXXXXX       XXXXXXX       X        99999
XXXXXXX       XXXXXXX       X        99999
--------                             ---------
Country                               999999
                                     =========
                                      9999999
```

subtotals

grand total

Generating Subtotals

Use a BY statement and a SUM statement in a PROC PRINT step.

```
proc sort data=orion.sales
          out=work.sales;
   by Country descending Salary;
run;

proc print data=work.sales noobs;
   by Country;
   sum Salary;
   var First_Name Last_Name Gender Salary;
run;
```

p104d10

Viewing the Output

```
----------------- Country=AU -----------------

First_Name    Last_Name      Gender    Salary

Tom           Zhou           M       108255
Wilson        Dawes          M        87975
Daniel        Pilgrim        M        36605
...
Kimiko        Tilley         F        25185
--------                     -------
Country                          1900015

----------------- Country=US -----------------

First_Name    Last_Name       Gender    Salary

Harry         Highpoint       M       243190
Louis         Favaron         M        95090
Dennis        Lansberry       M        84260
...
Tulsidas      Ould            M        22710
------------                 -------
Country                          3241405
                             =======
                             5141420
```

subtotal for AU

subtotal for US

grand total

Setup for the Poll

Modify the previous report to display only employees earning less than 25,500. Which WHERE statement (or statements) will result in the most efficient processing?

```
proc sort data=orion.sales
        out=work.sales;
   /* where Salary<25500;*/
   by Country descending Salary;
run;
proc print data=work.sales noobs;
   by Country;
   sum Salary;
   /*    where Salary<25500;*/
   var First_Name Last_Name Gender Salary;
run;
```

p104a03

4.07 Multiple Choice Poll

Which WHERE statement (or statements) will result in the most efficient processing?

- The WHERE statement in the PROC SORT step.
- The WHERE statement in the PROC PRINT step.
- Both WHERE statements are needed.
- The WHERE statements are equally efficient.

4.07 Multiple Choice Poll – Correct Answer

Which WHERE statement (statements) will result in the most efficient processing?

- The WHERE statement in the PROC SORT step.
- The WHERE statement in the PROC PRINT step.
- Both WHERE statements are needed.
- The WHERE statements are equally efficient.

Subsetting in the PROC SORT is more efficient. It selects and sorts only the required observations.

Be sure to use the OUT= option when subsetting in a PROC SORT or you will overwrite your original data set with the subset.

p104a03s

Exercise

This exercise reinforces the concepts discussed previously.

Chapter 4: Producing Detail Reports

Objectives

- Include titles and footnotes in a report.
- Define descriptive column headings using the LABEL statement.
- Control the use of column headings with the LABEL and SPLIT= options.

Business Scenario

Enhance the payroll report by adding titles, footnotes, and descriptive column headings.

Obs	Employee_ID	Last_ Name	Salary
1	9999	xxxxxxxxxx	99999
2	9999	xxxxxxxxxx	99999
3	99		

Orion Star Sales Staff
Salary Report

Obs	Employee ID	Last Name	Annual Salary
1	9999	xxxxxxxxxx	99999
2	9999	xxxxxxxxxx	99999
3	9999	xxxxxxxxxx	99999

Confidential

Displaying Titles and Footnotes

Use TITLE and FOOTNOTE statements to enhance the report.

TITLE*n* '*text* ';

FOOTNOTE*n* '*text* ';

```
title1 'Orion Star Sales Staff';
title2 'Salary Report';

footnote1 'Confidential';

proc print data=orion.sales;
   var Employee_ID Last_Name Salary;
run;

title;
footnote;
```

p104d11

Viewing the Output

Partial PROC PRINT Output

Orion Star Sales Staff
Salary Report

Obs	Employee_ID	Last_Name	Salary
1	120102	Zhou	108255
2	120103	Dawes	87975
3	120121	Elvish	26600
...			
164	121144	Capachietti	83505
165	121145	Lansberry	84260

Confidential

TITLE Statement

The global *TITLE statement* specifies title lines for SAS output.

TITLE*n* '*text* ';

- Titles appear at the top of the page.
- The default title is **The SAS System**.
- The value of ***n*** can be from 1 to 10.
- An unnumbered **TITLE** is equivalent to **TITLE1**.
- Titles remain in effect until they are changed or canceled, or you end your SAS session.

FOOTNOTE Statement

The global *FOOTNOTE statement* specifies footnote lines for SAS output.

FOOTNOTE*n* '*text* ';

- Footnotes appear at the bottom of the page.
- No footnote is printed unless one is specified.
- The value of ***n*** can be from 1 to 10.
- An unnumbered **FOOTNOTE** is equivalent to **FOOTNOTE1**.
- Footnotes remain in effect until they are changed or canceled, or you end your SAS session.

Changing Titles and Footnotes

To change a title line, submit a TITLE statement with the same number but different text.

- replaces a previous title with the same number
- cancels all titles with higher numbers

```
title1 'ABC Company';
title2 'Sales Division';
title3 'Salary Report';
```

```
title1 'Salary Report';
```

This statement changes title 1 and cancels titles 2 and 3.

Footnotes are changed the same way.

Canceling All Titles and Footnotes

- The null TITLE statement cancels all titles.

```
title;
```

- The null FOOTNOTE statement cancels all

```
footnote;
```

es.

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`title1 'The First Line';`
`title2 'The Second Line';`
`proc print data=orion.sales;`
`run;`	

`title2 'The Next Line';`
`proc print data=orion.sales;`
`run;`	

`title 'The Top Line';`
`proc print data=orion.sales;`
`run;`	

`title3 'The Third Line';`
`proc print data=orion.sales;`
`run;`	

`title;`
`proc print data=orion.sales;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`title1 'The First Line';`
`title2 'The Second Line';`
`proc print data=orion.sales;`
`run;`	

`title2 'The Next Line';`
`proc print data=orion.sales;`
`run;`	

`title 'The Top Line';`
`proc print data=orion.sales;`
`run;`	

`title3 'The Third Line';`
`proc print data=orion.sales;`
`run;`	

`title;`
`proc print data=orion.sales;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`title1 'The First Line';`
`title2 'The Second Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Second Line

`title2 'The Next Line';`
`proc print data=orion.sales;`
`run;`	

`title 'The Top Line';`
`proc print data=orion.sales;`
`run;`	

`title3 'The Third Line';`
`proc print data=orion.sales;`
`run;`	

`title;`
`proc print data=orion.sales;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`title1 'The First Line';`
`title2 'The Second Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Second Line

`title2 'The Next Line';`
`proc print data=orion.sales;`
`run;`	

`title 'The Top Line';`
`proc print data=orion.sales;`
`run;`	

`title3 'The Third Line';`
`proc print data=orion.sales;`
`run;`	

`title;`
`proc print data=orion.sales;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`title1 'The First Line';`
`title2 'The Second Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Second Line

`title2 'The Next Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Next Line

`title 'The Top Line';`
`proc print data=orion.sales;`
`run;`	

`title3 'The Third Line';`
`proc print data=orion.sales;`
`run;`	

`title;`
`proc print data=orion.sales;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`title1 'The First Line';`
`title2 'The Second Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Second Line

`title2 'The Next Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Next Line

`title 'The Top Line';`
`proc print data=orion.sales;`
`run;`	

`title3 'The Third Line';`
`proc print data=orion.sales;`
`run;`	

`title;`
`proc print data=orion.sales;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`title1 'The First Line';`
`title2 'The Second Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Second Line

`title2 'The Next Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Next Line

`title 'The Top Line';`
`proc print data=orion.sales;`
`run;`
	The Top Line

`title3 'The Third Line';`
`proc print data=orion.sales;`
`run;`	

`title;`
`proc print data=orion.sales;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`title1 'The First Line';`
`title2 'The Second Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Second Line

`title2 'The Next Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Next Line

`title 'The Top Line';`
`proc print data=orion.sales;`
`run;`
	The Top Line

`title3 'The Third Line';`
`proc print data=orion.sales;`
`run;`	

`title;`
`proc print data=orion.sales;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`title1 'The First Line';`
`title2 'The Second Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Second Line

`title2 'The Next Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Next Line

`title 'The Top Line';`
`proc print data=orion.sales;`
`run;`
	The Top Line

`title3 'The Third Line';`
`proc print data=orion.sales;`
`run;`
	The Top Line
	
	The Third Line

`title;`
`proc print data=orion.sales;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`title1 'The First Line';`
`title2 'The Second Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Second Line

`title2 'The Next Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Next Line

`title 'The Top Line';`
`proc print data=orion.sales;`
`run;`
	The Top Line

`title3 'The Third Line';`
`proc print data=orion.sales;`
`run;`
	The Top Line
	
	The Third Line

`title;`
`proc print data=orion.sales;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`title1 'The First Line';`
`title2 'The Second Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Second Line

`title2 'The Next Line';`
`proc print data=orion.sales;`
`run;`
	The First Line
	The Next Line

`title 'The Top Line';`
`proc print data=orion.sales;`
`run;`
	The Top Line

`title3 'The Third Line';`
`proc print data=orion.sales;`
`run;`
	The Top Line
	
	The Third Line

`title;`
`proc print data=orion.sales;`
`run;`	

4.08 Quiz

Which footnote or footnotes appear in the second procedure output?

a. Non Sales Employees

Non Sales Employees
Confidential

b. Orion Star
Non Sales Employees

Orion Star
Non Sales Employees
Confidential

```
footnote1 'Orion Star';
footnote2 'Sales Employees';
footnote3 'Confidential';
proc print data=orion.sales;
run;

footnote2 'Non Sales Employees';
proc print data=orion.nonsales;
run;
```

4.08 Quiz – Correct Answer

Which footnote or footnotes appear in the second procedure output?

a. Non Sales Employees

Non Sales Employees
Confidential

b. Orion Star
Non Sales Employees

Orion Star
Non Sales Employees
Confidential

```
footnote1 'Orion Star';
footnote2 'Sales Employees';
footnote3 'Confidential';
proc print data=orion.sales;
run;

footnote2 'Non Sales Employees';
proc print data=orion.nonsales;
run;
```

Idea Exchange

Which of the following programs do you prefer and why?

a.
```
title 'Orion Star Employees';
proc print data=orion.staff;
   where Gender='F';
   var Employee_ID Salary;
run;
```

b.
```
title 'Orion Star Female Employees';
proc print data=orion.staff;
   where Gender='F';
   var Employee_ID Salary;
run;
```

c.
```
title 'Orion Star Employees';
proc print data=orion.staff;
   where Gender='F';
   var Employee_ID Gender Salary;
run;
```

d.
```
title 'Orion Star Female Employees';
proc print data=orion.staff;
   where Gender='F';
   var Employee_ID Gender Salary;
run;
```

LABEL Statement and Option

Use a LABEL statement and the LABEL option to display descriptive column headings instead of variable names.

```
title1 'Orion Star Sales Staff';
title2 'Salary Report';
footnote1 'Confidential';

proc print data=orion.sales label;
   var Employee_ID Last_Name Salary;
   label Employee_ID='Sales ID'
       Last_Name='Last Name'
       Salary='Annual Salary';
run;

title;
footnote;
```

LABEL *variable-1*=**'***label***'**
   ...
      variable-n=**'***label***';**

p104d12

LABEL Statement

The LABEL statement assigns descriptive labels to variables.

- A label can be up to 256 characters and include any characters, including blanks.
- Labels are used automatically by many procedures.
- The PRINT procedure uses labels only when the LABEL or SPLIT= option is specified.

Viewing the Output

Orion Star Sales Staff
Salary Report

Obs	Sales ID	Last Name	Annual Salary
1	120102	Zhou	108255
2	120103	Dawes	87975
3	120121	Elvish	26600
...			
164	121144	Capachietti	83505
165	121145	Lansberry	84260

Confidential

SPLIT= Option

The SPLIT= option in PROC PRINT specifies a split character to control line breaks in column headings.

```
proc print data=orion.sales split='*';
   var Employee_ID Last_Name Salary;
   label Employee_ID='Sales ID'
         Last_Name='Last*Name'
         Salary='Annual*Salary';
run;
```

SPLIT='*split-character***'**

The SPLIT= option can be used instead of the LABEL option in a PROC PRINT step.

Viewing the Output

Partial PROC PRINT Output

Orion Star Sales Staff
Salary Report

Obs	Last Sales ID	Name	Annual Salary
1	120102	Zhou	108255
2	120103	Dawes	87975
3	120121	Elvish	26600
...			
164	121144	Capachietti	83505
165	121145	Lansberry	84260

Confidential

Exercise

This exercise reinforces the concepts discussed previously.

Chapter Review

1. Which observation or observations will be selected by the following WHERE statement?

```
where Job_Title contains 'I';
```

Obs	Last_Name	First_Name	Country	Job_Title
1	Wu	Christine	AU	Sales Rep. I
2	Stone	Kimiko	AU	Sales Manager
3	Hofmann	Fred	AU	Insurance Sales

- observation 1
- observation 2
- observation 3
- observations 1 and 3
- all observations

1. Which observation or observations will be selected by the following WHERE statement?

```
where Job_Title contains 'I';
```

Obs	Last_Name	First_Name	Country	Job_Title
1	Wu	Christine	AU	Sales Rep. I
2	Stone	Kimiko	AU	Sales Manager
3	Hofmann	Fred	AU	Insurance Sales

- observation 1
- observation 2
- observation 3
- observations 1 and 3
- all observations

2. Which statement in a PROC SORT step prepares data to be displayed as shown in this output?

Postal_Code	Employee_ID
92129	121074
92129	121001
92128	121128
92128	120755
92128	120730

a. `by Postal_Code Employee_ID;`

b. `by descending Postal_Code Employee_ID;`

c. `by Postal_Code descending Employee_ID;`

d. `by descending Postal_Code descending Employee_ID;`

2. Which statement in a PROC SORT step prepares data to be displayed as shown in this output?

Postal_Code	Employee_ID
92129	121074
92129	121001
92128	121128
92128	120755
92128	120730

a. `by Postal_Code Employee_ID;`

b. `by descending Postal_Code Employee_ID;`

c. `by Postal_Code descending Employee_ID;`

d. `by descending Postal_Code descending Employee_ID;`

3. Which statement about this PROC SORT step is true?

```
proc sort data=orion.staff;
          out=work.staff;
   by descending Salary
      Manager_ID;
run;
```

- The sorted data set overwrites the input data set.
- The observations are sorted by **Salary** in descending order, and then by **Manager_ID** in descending order.
- A semicolon should not appear after the input data set name.
- The sorted data set contains only the variables specified in the BY statement.

3. Which statement about this PROC SORT step is true?

```
proc sort data=orion.staff;
          out=work.staff;
   by descending Salary
      Manager_ID;
run;
```

- The sorted data set overwrites the input data set.
- The observations are sorted by **Salary** in descending order, and then by **Manager_ID** in descending order.
- A semicolon should not appear after the input data set name.
- The sorted data set contains only the variables specified in the BY statement.

4. Which of the following statements selects from a data set only those observations for which the value of the variable **Style** is *RANCH*, *SPLIT*, or *TWOSTORY*?

a. `where Style='RANCH' or 'SPLIT' or 'TWOSTORY';`

b. `where Style in 'RANCH' or 'SPLIT' or 'TWOSTORY';`

c. `where Style in (RANCH, SPLIT, TWOSTORY);`

d. `where Style in ('RANCH','SPLIT','TWOSTORY');`

4. Which of the following statements selects from a data set only those observations for which the value of the variable **Style** is *RANCH*, *SPLIT*, or *TWOSTORY*?

a. `where Style='RANCH' or 'SPLIT' or 'TWOSTORY';`

b. `where Style in 'RANCH' or 'SPLIT' or 'TWOSTORY';`

c. `where Style in (RANCH, SPLIT, TWOSTORY);`

d. `where Style in ('RANCH','SPLIT','TWOSTORY');`

5. Which of the following statements selects rows in which **Amount** is less than or equal to $5,000 or **Rate** equals 0.095?

   a. `where Amount<=5000 or Rate=0.095;`
   b. `where Amount le 5000 or Rate=0.095;`
   c. `where Amount<=5000 or Rate eq 0.095;`
   d. all of the above

5. Which of the following statements selects rows in which **Amount** is less than or equal to $5,000 or **Rate** equals 0.095?

a. `where Amount<=5000 or Rate=0.095;`
b. `where Amount le 5000 or Rate=0.095;`
c. `where Amount<=5000 or Rate eq 0.095;`
d. all of the above

6. When you run this code, which title or titles appear in the last PROC PRINT output?

   - The Top Line
   - The Top Line
     The Next Line
   - The Top Line
     The First Line
     The Next Line

```
title1 'The First Line';
title2 'The Second Line';
proc print data=orion.sales;
run;

title2 'The Next Line';
proc print data=orion.sales;
run;

title 'The Top Line';
proc print data=orion.sales;
run;
```

6. When you run this code, which title or titles appear in the last PROC PRINT output?

- The Top Line
- The Top Line
  The Next Line
- The Top Line
  The First Line
  The Next Line

```
title1 'The First Line';
title2 'The Second Line';
proc print data=orion.sales;
run;

title2 'The Next Line';
proc print data=orion.sales;
run;

title 'The Top Line';
proc print data=orion.sales;
run;
```

7. Which program creates the output shown here?

a.
```
proc print data=orion.staff;
   var Employee_ID Emp_Hire_Date;
   label Employee_ID='Emp ID'
         'Employee_Hire Date';
run;
```

b.
```
proc print data=orion.staff split='+';
   var Employee_ID Emp_Hire_Date;
   label Employee_ID='Emp ID'
         Emp_Hire_Date='Employee+Hire Date';
run;
```

Partial PROC PRINT Output

Obs	Emp ID	Employee Hire Date
1	120101	01JUL2003
2	120102	01JUN1989
3	120103	01JAN1974
4	120104	01JAN1981
5	120105	01MAY1999
6	120106	01JAN1974

7. Which program creates the output shown here?

a.
```
proc print data=orion.staff;
   var Employee_ID Emp_Hire_Date;
   label Employee_ID='Emp ID'
         'Employee_Hire Date';
run;
```

b.
```
proc print data=orion.staff split='+';
   var Employee_ID Emp_Hire_Date;
   label Employee_ID='Emp ID'
         Emp_Hire_Date='Employee+Hire Date';
run;
```

Partial PROC PRINT Output

Obs	Emp ID	Employee Hire Date
1	120101	01JUL2003
2	120102	01JUN1989
3	120103	01JAN1974
4	120104	01JAN1981
5	120105	01MAY1999
6	120106	01JAN1974

8. Which BY statement is valid for this PROC PRINT step?

```
proc sort data=orion.staff
          out=work.staffsort;
   by Gender Start_Date;
run;

proc print data=work.staffsort label;
   by ___________________________;
   label Start_Date='Start';
run;
```

a. `by Start_Date Gender;`

b. `by Start;`

c. `by descending Gender;`

d. `by Gender;`

8. Which BY statement is valid for this PROC PRINT step?

```
proc sort data=orion.staff
          out=work.staffsort;
   by Gender Start_Date;
run;

proc print data=work.staffsort label;
   by ____________________________;
   label Start_Date='Start';
run;
```

a. `by Start_Date Gender;`

b. `by Start;`

c. `by descending Gender;`

d. `by Gender;`

9. Suppose you already ran the first program, which created a one-page report. Next, you want to run the second program. What will appear at the top of the second report?

```
title1 'RADIX Company';
title3 'DVD Sales';
proc print data=radix.sales;
   where UnitSold>=30;
run;
```

```
title2 'Best Sales';
title;
proc print data=radix.staff;
   where Sales>25000;
run;
```

- no titles
- RADIX Company
  Best Sales
  DVD Sales
- RADIX Company
  Best Sales
- RADIX Company

9. Suppose you already ran the first program, which created a one-page report. Next, you want to run the second program. What will appear at the top of the second report?

```
title1 'RADIX Company';
title3 'DVD Sales';
proc print data=radix.sales;
   where UnitSold>=30;
run;
```

```
title2 'Best Sales';
title;
proc print data=radix.staff;
   where Sales>25000;
run;
```

- no titles
- RADIX Company
  Best Sales
  DVD Sales
- RADIX Company
  Best Sales
- RADIX Company

10. Which statement about this program is true?

```
proc print data=orion.sales;
   var Employee_ID Salary;
   where Country='AU';
   by Gender;
   label Salary='Annual Salary';
run;
```

- This program will run correctly only if **orion.sales** is sorted in ascending order by **Country**.
- The PROC PRINT report displays only the observations in which the value of **Country** is *AU*.
- *Annual Salary* will be displayed at the top of the **Salary** column.

10. Which statement about this program is true?

```
proc print data=orion.sales;
   var Employee_ID Salary;
   where Country='AU';
   by Gender;
   label Salary='Annual Salary';
run;
```

- This program will run correctly only if **orion.sales** is sorted in ascending order by **Country**.
- The PROC PRINT report displays only the observations in which the value of **Country** is *AU*.
- *Annual Salary* will be displayed at the top of the **Salary** column.