UCD School of Mathematics and Statistics

STAT40840: Data programming with SAS
Laura Kirwan

Lecture 5

www.ucd.ie/mathstat

1 Reading a SAS Data Set

2 Customizing a SAS Data Set

Objectives

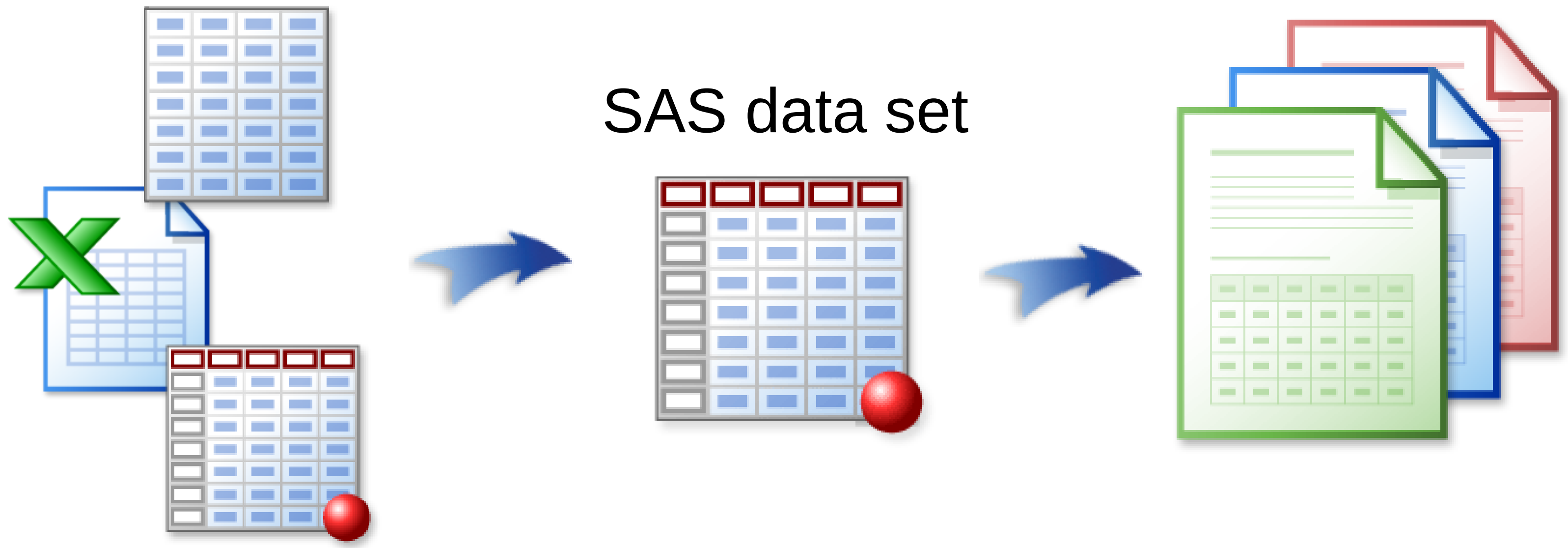
- Define the scenario that is used when you read from a data source to create a SAS data set.
- Use a DATA step to create a SAS data set from an existing SAS data set.
- Subset observations with a WHERE statement.
- Create a new variable with an assignment statement.

Scenario

Information about Orion Star sales employees resides in several input sources.

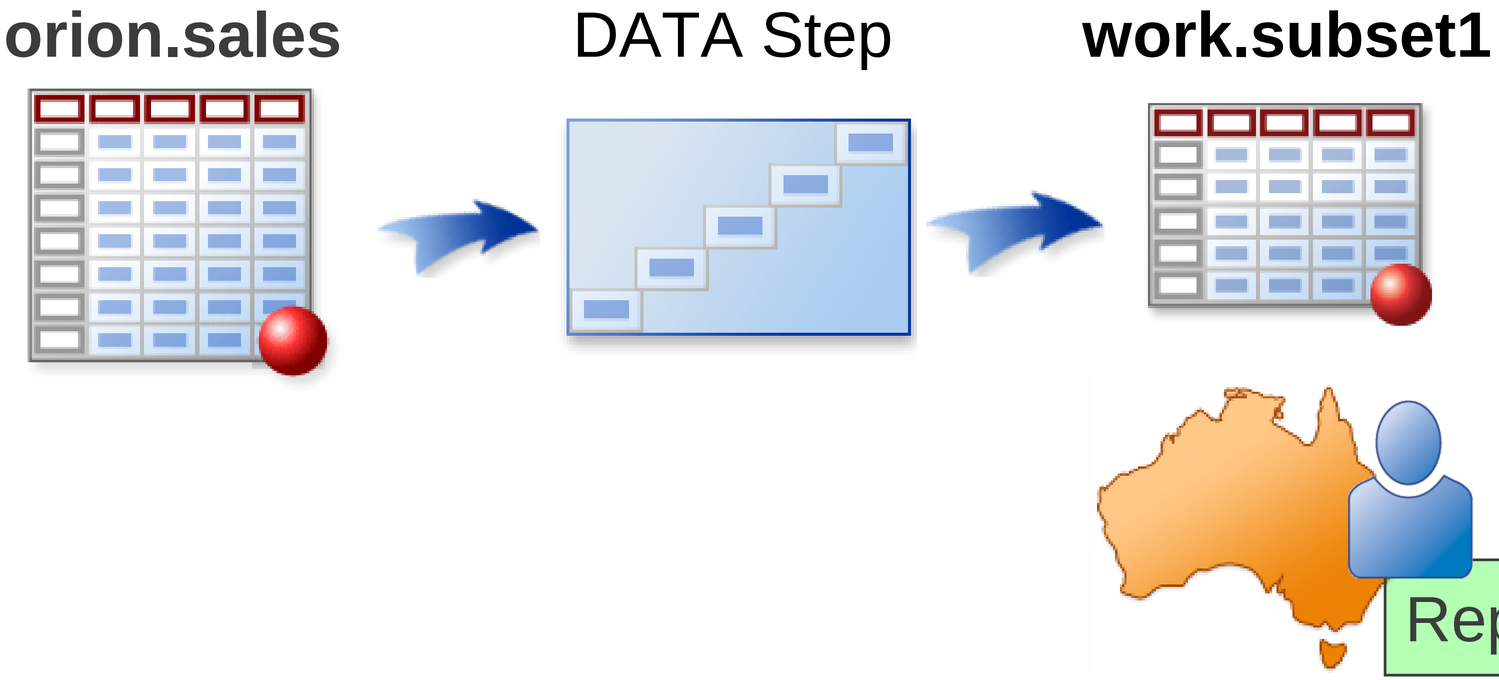
Considerations

Management wants a series of reports for Australian sales employees. You read data from various input sources to create a SAS data set that can be analyzed and presented.



Scenario: Part 1

Read an existing SAS data set to create a new data set. The new data set should include only the observations
for the Australian sales representatives.



Using a SAS Data Set as Input

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
run;
```

DATA *output-SAS-data-set***;**
  SET *input-SAS-data-set***;**
  WHERE *WHERE-expression***;**
RUN;

L5_D1.sas

DATA Statement

The *DATA statement* begins a DATA step and provides the name of the SAS data set to create.

DATA *output-SAS-data-set*;

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
run;
```

A DATA step can create temporary or permanent data sets.

L5_D1.sas

✎ The rules for SAS variable names also apply to data set names.

SET Statement

The *SET statement* reads observations from an existing SAS data set for further processing in the DATA step.

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
run;
```

SET *input-SAS-data-set*;

- The SET statement reads all observations and all variables from the input data set.
- Observations are read sequentially, one at a time.
- The SET statement can read temporary or permanent data sets.

L5_D1.sas

WHERE Statement

The *WHERE statement* selects observations from a SAS data set that meet a particular condition.

WHERE *WHERE-expression*;

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
run;
```

The variables named in the WHERE expression must exist in the input SAS data set.

L5_D1.sas

Viewing the Log

```
42   data work.subset1;
43      set orion.sales;
44      where Country='AU' and
45            Job_Title contains 'Rep';
46   run;

NOTE: There were 61 observations read from the data set ORION.SALES.
      WHERE (Country='AU') and Job_Title contains 'Rep';
NOTE: The data set WORK.SUBSET1 has 61 observations and 9 variables.
```

- SAS read 61 of the 165 observations.

Viewing the Output

```
proc print data=work.subset1 noobs;
run;
```

Partial PROC PRINT Output

```
              First_                                                               Birth_    Hire_
Employee_ID   Name       Last_Name   Gender    Salary  Job_Title        Country     Date      Date

     120121   Irenie     Elvish        F        26600  Sales Rep. II      AU        -4169     6575
     120122   Christina  Ngan          F        27475  Sales Rep. II      AU         -523     8217
     120123   Kimiko     Hotstone      F        26190  Sales Rep. I       AU         3193    10866
     120124   Lucian     Daymond       M        26480  Sales Rep. I       AU         1228     8460
     120125   Fong       Hofmeister    M        32040  Sales Rep. IV      AU         -391     8460
```

L5_D1.sas

Exercise 1

Consider the DATA step below.

```
data us;
   set orion.sales;
   where Country='US';
run;
```

L5_E1.sas

Exercise 1

Considering this DATA step, which statement is true?

a. It reads a temporary data set and creates a permanent data set.

b. It reads a permanent data set and creates a temporary data set.

c. It contains a syntax error and does not execute.

d. It does not execute because you cannot work with permanent and temporary data sets in the same step.

Exercise 1 solution

- Considering this DATA step, which statement is true?

a. It reads a temporary data set and creates a permanent data set.

[✓ b.] It reads a permanent data set and creates a temporary data set.

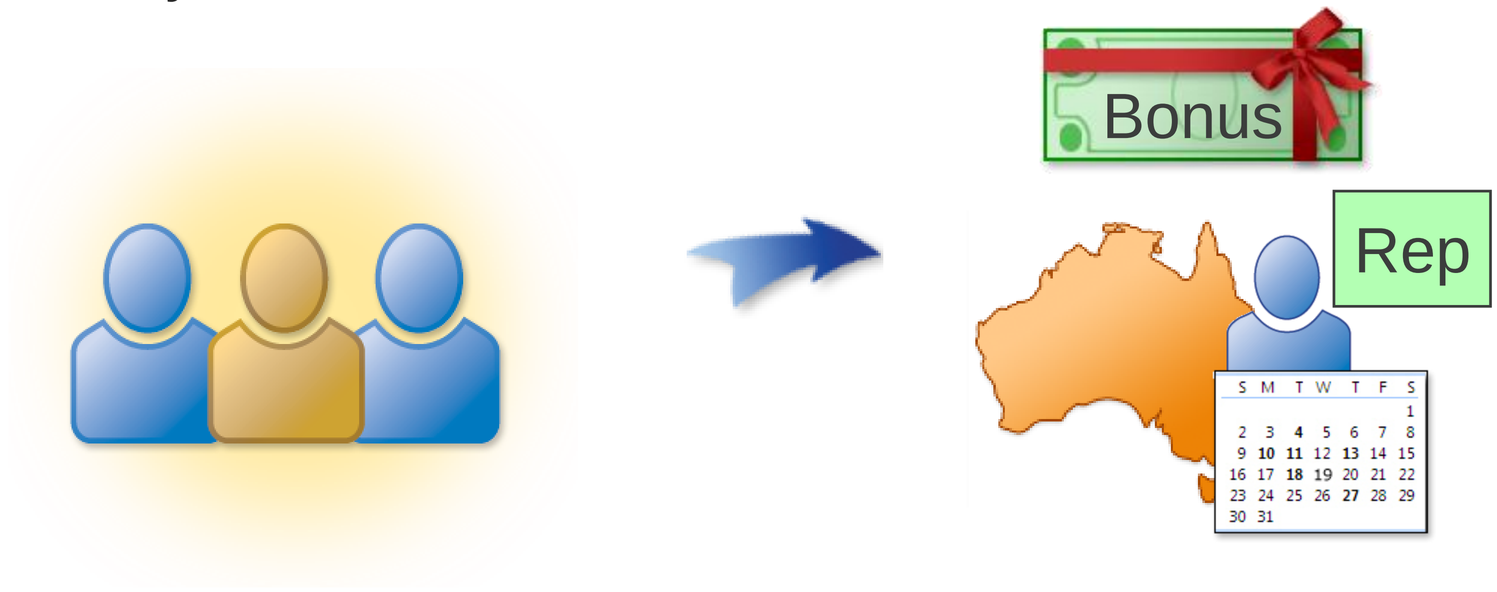
c. It contains a syntax error and does not execute.

d. It does not execute because you cannot work with permanent and temporary data sets in the same step.

```
data us;                  /* Create a temporary data set */
   set orion.sales;        /* Read a permanent data set */
   where Country='US';
run;
```

Scenario: Part 2

Orion Star management wants to give a 10% bonus to each Australian Sales representative hired before January 1, 2000.



Considerations

Subsetting is based on **Hire_Date**, which contains a SAS date value. How can you compare a SAS date value to a calendar date?

Use a SAS date constant.

Date Constant

A date constant can be used in any SAS expression, including a WHERE expression.

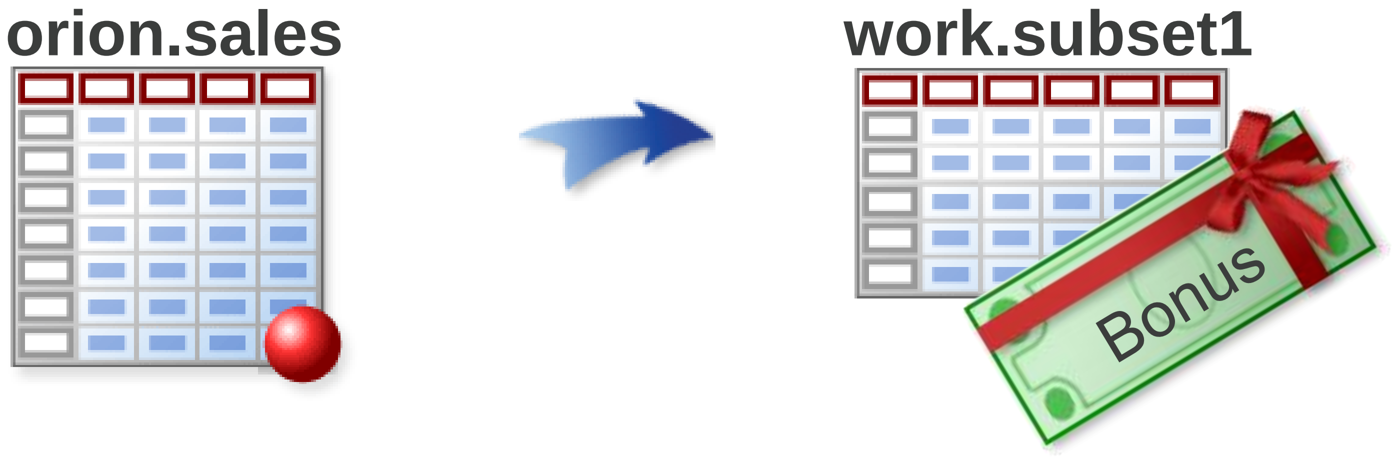
```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep' and
         Hire_Date<'01jan2000'd;
run;
```

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

L5_D2.sas

Considerations

Create a data set that includes the new variable, **Bonus**, which represents a 10% bonus.



Assignment Statement

The *assignment statement* evaluates an expression and assigns the result to a new or existing variable.

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep' and
         Hire_Date<'01jan2000'd;
   Bonus=Salary*.10;
run;
```

***variable*=*expression*;**

L5_D2a.sas

Assignment Statement

The *expression* consists of operands and operators.

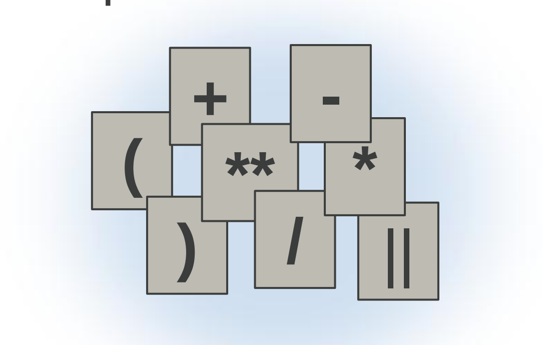
***variable=expression*;**

Operands

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

Operators

- symbols that represent a calculation or manipulation



- SAS functions

Sample Assignment Statements

Example	Type
`Salary=26960;`	Numeric constant
`Gender='F';`	Character constant
`Hire_Date='21JAN1995'd;`	Date constant
`BonusMonth=month(Hire_Date);`	SAS function
`Bonus=Salary*.10;`	Arithmetic expression

Arithmetic Operators

If any operand in an arithmetic expression has a missing value, the result is a missing value.

Symbol	Definition	Priority
**	Exponentiation	I
*	Multiplication	II
/	Division	II
+	Addition	III
-	Subtraction	III

Parentheses can be used to clarify or alter the order of operations in an arithmetic expression.

Viewing the Log

```
214  data work.subset1;
215     set orion.sales;
216     where Country='AU' and
217           Job_Title contains 'Rep' and
218           Hire_Date<'01jan2000'd;
219     Bonus=Salary*.10;
220  run;

NOTE: There were 29 observations read from the data set ORION.SALES.
      WHERE (Country='AU') and Job_Title contains 'Rep' and
      (Hire_Date<'01JAN2000'D);
NOTE: The data set WORK.SUBSET1 has 29 observations and 10 variables.
```

The input data set has 9 variables, and the new data set has 10 variables.

Viewing the Output

```
proc print data=work.subset1 noobs;
   var First_Name Last_Name Salary
       Job_Title Bonus Hire_Date;
   format Hire_Date date9.;
run;
```

Partial PROC PRINT Output

L5_D2a.sas

```
First_Name    Last_Name    Salary     Job_Title       Bonus     Hire_Date

Irenie        Elvish        26600   Sales Rep. II    2660.0     01JAN1978
Christina     Ngan          27475   Sales Rep. II    2747.5     01JUL1982
Kimiko        Hotstone      26190   Sales Rep. I     2619.0     01OCT1989
Lucian        Daymond       26480   Sales Rep. I     2648.0     01MAR1983
Fong          Hofmeister    32040   Sales Rep. IV    3204.0     01MAR1983
```

Exercise 2

What are the values of **n1** and **n2** given the following variables and values?

x	y	z
.	4	10

a. `n1=y+z/2;`

b. `n2=x+z/2;`

Exercise 2 solution

What are the values of **n1** and **n2** given the following variables and values?

x	y	z
.	4	10

a. $\boxed{\texttt{n1=y+z/2;}} \longrightarrow 4+10/2 \longrightarrow 4+5 \longrightarrow \bigcirc\!\!\!\!9$

$\boxed{\texttt{n1=(y+z)/2;}} \longrightarrow 14/2 \longrightarrow 7$

b. $\boxed{\texttt{n2=x+z/2;}} \longrightarrow .+10/2 \longrightarrow .+5 \longrightarrow \bigcirc\!\!\!\!.$

Lecture 5 Part 2

1 Reading a SAS Data Set

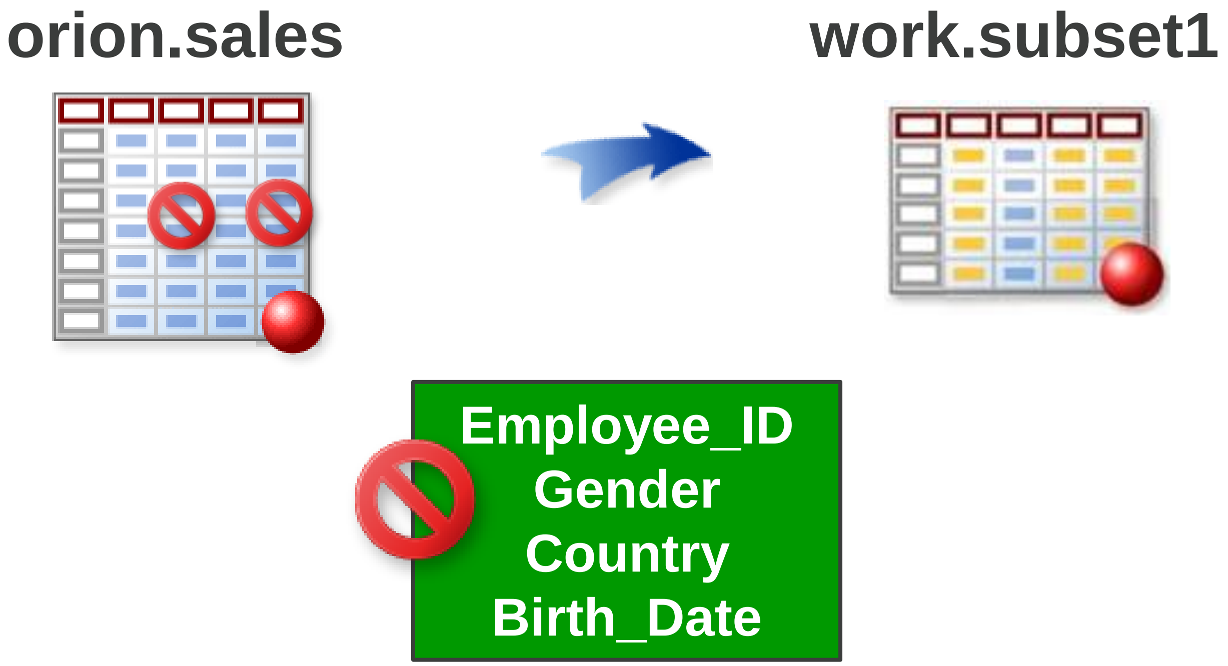
2 Customizing a SAS Data Set

Objectives

- Subset variables by using the DROP and KEEP statements.
- Explore the compilation and execution phases of the DATA step.
- Store labels and formats in the descriptor portion of a SAS data set.

Scenario: Part 3

All Australian sales representatives receive a bonus, regardless of hire date. The new data set should contain a subset of the variables from the input data set.



DROP Statement

The DROP statement specifies the variables to ***exclude*** from the output data set.

L5_D3.sas

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

DROP *variable-list*;

```
NOTE: There were 61 observations read from the data set ORION.SALES.
      WHERE (Country='AU') and Job_Title contains 'Rep';
NOTE: The data set WORK.SUBSET1 has 61 observations and 6 variables.
```

Viewing the Output

```
proc print data=work.subset1;
run;
```

Partial PROC PRINT Output

```
     First_                                                   Hire_
Obs  Name          Last_Name       Salary   Job_Title          Date      Bonus

1    Irenie        Elvish           26600   Sales Rep. II      6575     2660.0
2    Christina     Ngan             27475   Sales Rep. II      8217     2747.5
3    Kimiko        Hotstone         26190   Sales Rep. I      10866     2619.0
4    Lucian        Daymond          26480   Sales Rep. I       8460     2648.0
5    Fong          Hofmeister       32040   Sales Rep. IV      8460     3204.0
```

L5_D3.sas

KEEP Statement

The KEEP statement specifies all variables to ***include*** in the output data set.

L5_D3a.sas

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   keep First_Name Last_Name Salary
        Job_Title Hire_Date Bonus;
run;
```

KEEP *variable-list***;**

If a KEEP statement is used, it must include ***every*** variable to be written, including any new variables.

Viewing the Log

Partial SAS Log

```
NOTE: There were 61 observations read from the data set ORION.SALES.
      WHERE (Country='AU') and Job_Title contains 'Rep';
NOTE: The data set WORK.SUBSET1 has 61 observations and 6 variables.
```

Viewing the Output

```
proc print data=work.subset1;
run;
```

Partial PROC PRINT Output

```
     First_                                                     Hire_
Obs  Name       Last_Name    Salary   Job_Title                 Date      Bonus

1    Irenie     Elvish        26600   Sales Rep. II             6575     2660.0
2    Christina  Ngan          27475   Sales Rep. II             8217     2747.5
3    Kimiko     Hotstone      26190   Sales Rep. I             10866     2619.0
4    Lucian     Daymond       26480   Sales Rep. I              8460     2648.0
5    Fong       Hofmeister    32040   Sales Rep. IV             8460     3204.0
```

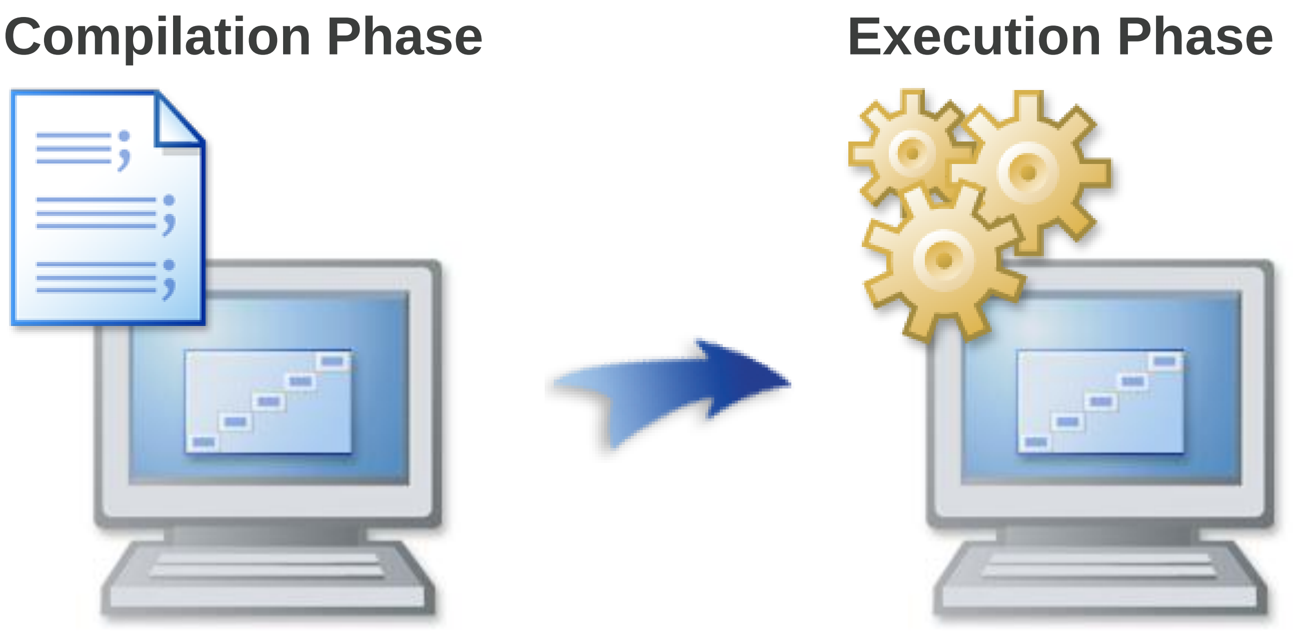
L5_D3a.sas

Scenario: Behind the Scenes

Orion Star programmers need to understand the internal processing that occurs when a DATA step is submitted.

DATA Step Processing

SAS processes the DATA step in two phases.



Compilation Phase

Scans the program for syntax errors; translates the program into machine language.

PDV

Name	Salary

Creates the *program data vector* (*PDV*) to hold one observation.

Creates the descriptor portion of the output data set.

Compilation

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

L5_D3.sas

Compilation

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

PDV

Employee_ID N 8	First_Name $ 12	Last_Name $ 18	Gender $ 1	Salary N 8	Job_Title $ 25

Country $ 2	Birth_Date N 8	Hire_Date N 8

Compilation

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

PDV

Employee_ID N 8	First_Name $ 12	Last_Name $ 18	Gender $ 1	Salary N 8	Job_Title $ 25

Country $ 2	Birth_Date N 8	Hire_Date N 8	Bonus N 8

Compilation

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

PDV

Employee_ID	First_Name	Last_Name	Gender	Salary	Job_Title
D N 8	$ 12	$ 18	D $ 1	N 8	$ 25

Country	Birth_Date	Hire_Date	Bonus
D $ 2	D N 8	N 8	N 8

...

Compilation

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

PDV

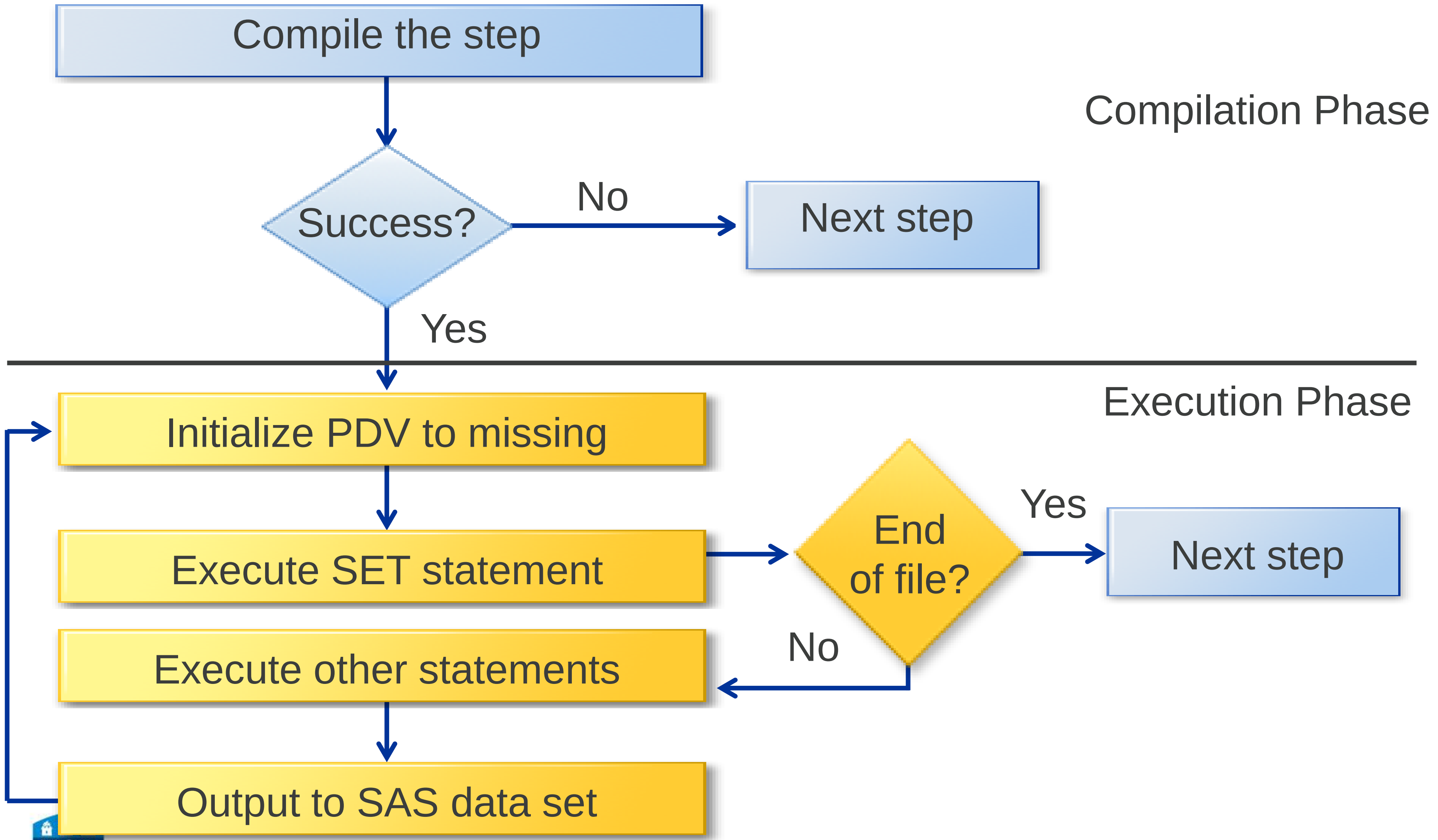
Employee_ID	First_Name	Last_Name	Gender	Salary	Job_Title
D N 8	$ 12	$ 18	D $ 1	N 8	$ 25

Country	Birth_Date	Hire_Date	Bonus
D $ 2	D N 8	N 8	N 8

Descriptor Portion of **work.subset1**

First_Name	Last_Name	Salary	Job_Title	Hire_Date	Bonus

Execution Phase



Execution

Initialize PDV

Partial **orion.sales**

Employee_ID	...	Hire_Date
120121		6575
120122	...	8217
120123		10866
120124		8460

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

PDV

D Employee_ID	...	Salary	...	D Country	D Birth_Date	Hire_Date	Bonus
.		.			.	.	.

work.subset1

First_Name	Last_Name	Salary	Job_Title	Hire_Date	Bonus

Execution

Partial **orion.sales**

Employee_ID	...	Hire_Date
120121		6575
120122	...	8217
120123		10866
120124		8460

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

PDV

D Employee_ID	...	Salary	...	D Country	D Birth_Date	Hire_Date	Bonus
120121		26600		AU	-4169	6575	.

work.subset1

First_Name	Last_Name	Salary	Job_Title	Hire_Date	Bonus

Execution

Partial **orion.sales**

Employee_ID	...	Hire_Date
120121		6575
120122	...	8217
120123		10866
120124		8460

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

PDV

D Employee_ID	...	Salary	...	D Country	D Birth_Date	Hire_Date	Bonus
120121		26600		AU	-4169	6575	2660

work.subset1

First_Name	Last_Name	Salary	Job_Title	Hire_Date	Bonus

Execution

Partial **orion.sales**

Employee _ID	...	Hire_Date
120121		6575
120122	...	8217
120123		10866
120124		8460

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

Employee _ID D	...	Salary	...	Country D	Birth_Date D	Hire_Date	Bonus
120121		26600		AU	-4169	6575	2660

work.subset1

First_Name	Last_Name	Salary	Job_Title	Hire_Date	Bonus
Irenie	Elvish	26600	Sales Rep. II	6575	2660

Execution

Reinitialize PDV

Partial **orion.sales**

Employee_ID	...	Hire_Date
120121		6575
120122		8217
120123		10866
120124		8460

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

New variables are reinitialized.

PDV

Employee_ID D	...	Salary	...	Country D	Birth_Date D	Hire_Date	Bonus
120121		26600		AU	-4169	6575	.

work.subset1

First_Name	Last_Name	Salary	Job_Title	Hire_Date	Bonus
Irenie	Elvish	26600	Sales Rep. II	6575	2660

Execution

Partial **orion.sales**

Employee_ID	...	Hire_Date
120121		6575
120122	...	8217
120123		10866
120124		8460

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

PDV

D Employee_ID	...	Salary	...	D Country	D Birth_Date	Hire_Date	Bonus
120122		27475		AU	-523	8217	.

work.subset1

First_Name	Last_Name	Salary	Job_Title	Hire_Date	Bonus
Irenie	Elvish	26600	Sales Rep. II	6575	2660

Execution

Partial **orion.sales**

Employee_ID
120121
120122
120123
120124

...

Hire_Date
6575
8217
10866
8460

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

PDV

D Employee_ID	...	Salary	...	D Country	D Birth_Date	Hire_Date	Bonus
120122		27475		AU	-523	8217	2747.5

work.subset1

First_Name	Last_Name	Salary	Job_Title	Hire_Date	Bonus
Irenie	Elvish	26600	Sales Rep. II	6575	2660

Execution

Partial **orion.sales**

Employee _ID	...	Hire_Date
120121		6575
120122	...	8217
120123		10866
120124		8460

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_D
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

D Employee _ID	...	Salary	...	D Country	D Birth_Date	Hire_Date	Bonus
120122		27475		AU	-523	8217	2747.5

work.subset1

First_Name	Last_Name	Salary	Job_Title	Hire_Date	Bonus
Irenie	Elvish	26600	Sales Rep. II	6575	2660.0
Christina	Ngan	27475	Sales Rep. II	8217	2747.5

...

Execution

Partial **orion.sales**

Employee _ID		Hire_Date
120121		6575
120122	...	8217
120123		10866
120124		8460

```
data work.subset1;
   set orion.sale
   where Country=
         Job_Titl
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

Continue until EOF

PDV

D Employee _ID	...	Salary	...	D Country	D Birth_Date	Hire_Date	Bonus
120122		27475		AU	-523	8217	.

work.subset1

First_Name	Last_Name	Salary	Job_Title	Hire_Date	Bonus
Irenie	Elvish	26600	Sales Rep. II	6575	2660.0
Christina	Ngan	27475	Sales Rep. II	8217	2747.5

Viewing the Output

```
proc print data=work.subset1;
run;
```

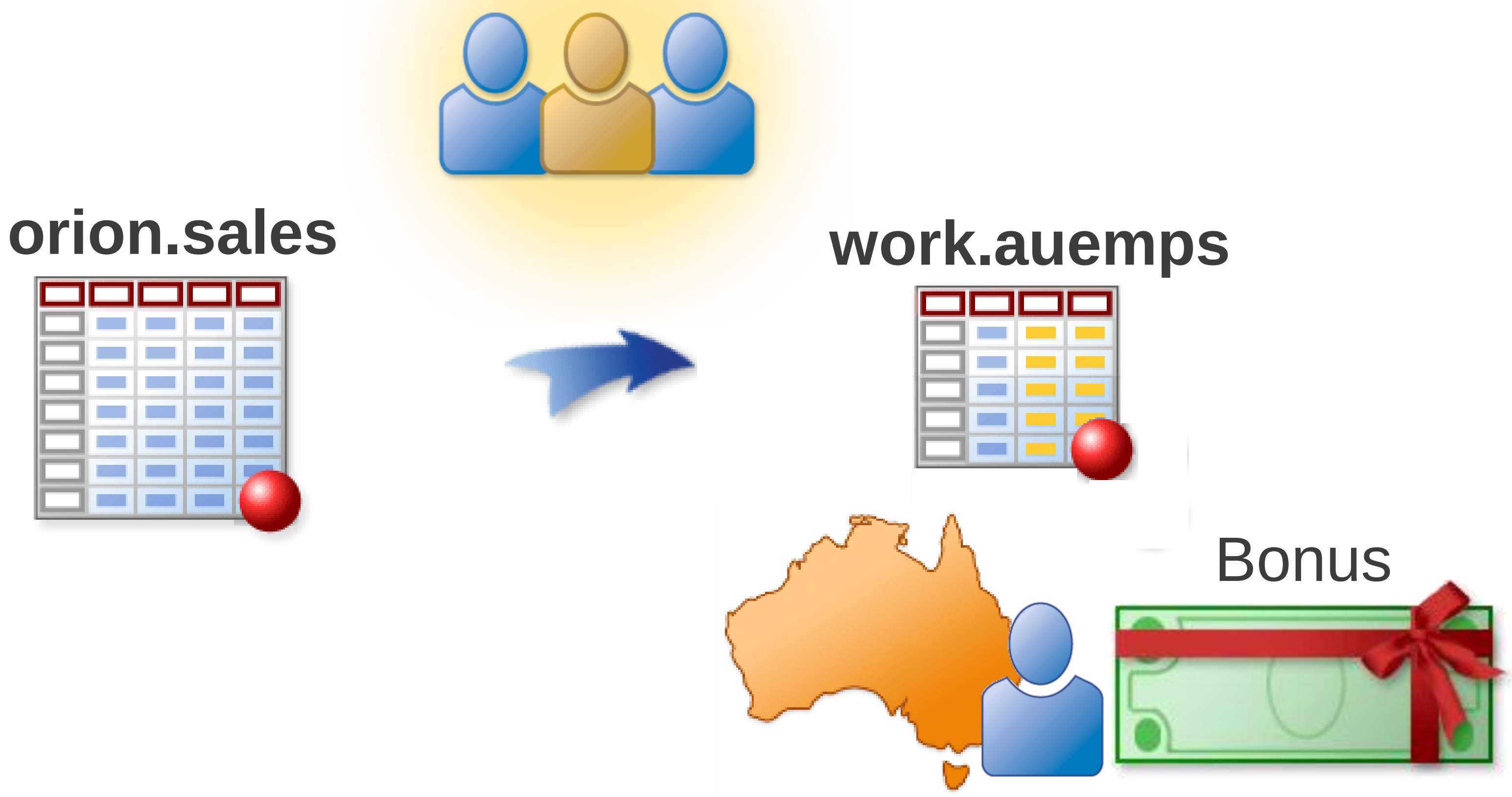
```
        First_                                                       Hire_
Obs     Name         Last_Name      Salary    Job_Title              Date       Bonus

  1     Irenie       Elvish          26600    Sales Rep. II          6575      2660.0
  2     Christina    Ngan            27475    Sales Rep. II          8217      2747.5
  3     Kimiko       Hotstone        26190    Sales Rep. I          10866      2619.0
  4     Lucian       Daymond         26480    Sales Rep. I           8460      2648.0
  5     Fong         Hofmeister      32040    Sales Rep. IV          8460      3204.0
```

L5_D3.sas

Scenario: Part 4

Create a data set that contains all Australian employees whose bonus is at least $3000.



Selecting Observations

Subsetting is based on the new variable, **Bonus**, that is created with an assignment statement.

```
data work.auemps;
   set orion.sales;
   where Country='AU';
   Bonus=Salary*.10;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

A WHERE statement is used to subset observations when the selected variables exist in the ***input*** data set.

L5_D4.sas

Exercise 3

Open and submit **L5_E3.sas**. Is the output data set created successfully?

```
data work.usemps;
   set orion.sales;
   Bonus=Salary*.10;
   where Country='US' and Bonus>=3000;
run;
```

L5_E3.sas

Exercise 3 solution

Open and submit **L5_E3.sas**. Is the output data set created successfully?

L5_E3.sas

```
260  data work.usemps;
261     set orion.sales;
262     Bonus=Salary*.10;
263     where Country='US' and Bonus>=3000;
ERROR: Variable Bonus is not on file ORION.SALES.
264  run;

NOTE: The SAS System stopped processing this step because of
errors.
WARNING: The data set WORK.USEMPS may be incomplete.  When
this step was stopped there were 0 observations and 10
variables.
```

No. Bonus cannot be used in a WHERE statement because it is not in the input data set. It is a new variable that is created in this DATA step.

Subsetting IF

The *subsetting IF* statement tests a condition to determine whether the DATA step should continue processing the current observation.

L5_D5.sas

```
data work.auemps;
   set orion.sales;
   where Country='AU';
   Bonus=Salary*.10;
   if Bonus>=3000;
run;
```

IF *condition*;

In this program, processing reaches the bottom of the DATA step and outputs an observation only if the condition is true.

Viewing the Log

Partial SAS Log

```
11    data work.auemps;
12       set orion.sales;
13       where Country='AU';
14       Bonus=Salary*.10;
15       if Bonus>=3000;
16    run;

NOTE: There were 63 observations read from the data set ORION.SALES.
      WHERE Country='AU';
NOTE: The data set WORK.AUEMPS has 12 observations and 10 variables.
```

Of the 165 observations in **orion.sales**, 63 were read into the PDV for processing, and only 12 were written to the new data set.

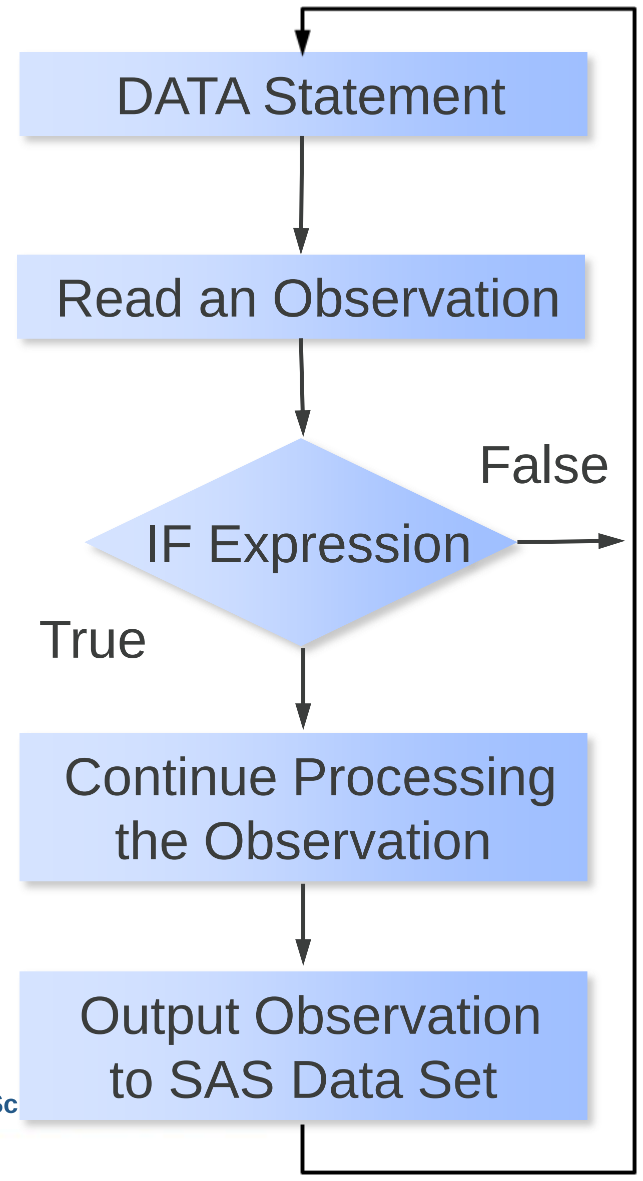
Viewing the Output

```
proc print data=work.auemps;
   var First_Name Last_Name Salary Bonus;
run;
```

```
       First_
Obs    Name        Last_Name       Salary      Bonus

  1    Tom         Zhou            108255    10825.5
  2    Wilson      Dawes            87975     8797.5
  3    Fong        Hofmeister       32040     3204.0
  4    Monica      Kletschkus       30890     3089.0
  5    Alvin       Roebuck          30070     3007.0
  6    Alexei      Platts           32490     3249.0
  7    Viney       Barbis           30265     3026.5
  8    Caterina    Hayawardhana     30490     3049.0
  9    Daniel      Pilgrim          36605     3660.5
 10    Lynelle     Phoumirath       30765     3076.5
 11    Rosette     Martines         30785     3078.5
 12    Fadi        Nowd             30660     3066.0
```

Processing the Subsetting IF Statement

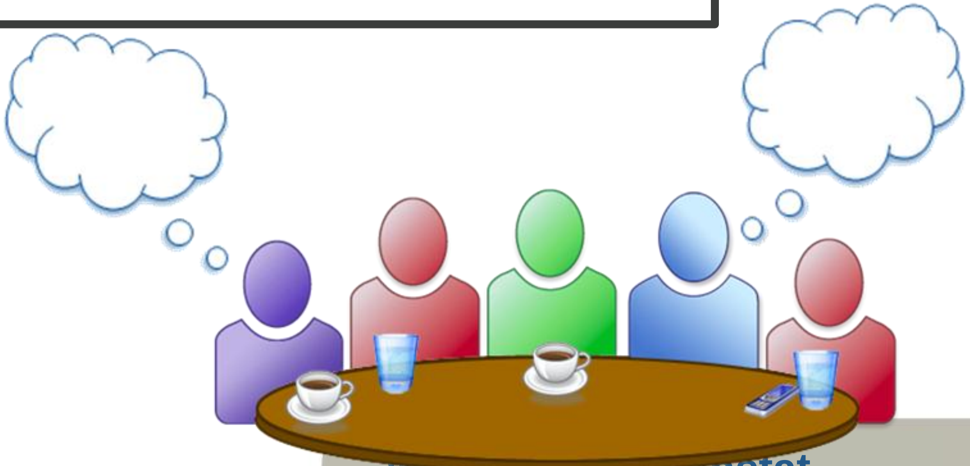

DATA Statement
Read an Observation
IF Expression
False
True
Continue Processing the Observation
Output Observation to SAS Data Set

A subsetting IF statement is valid only in a DATA step.

Exercise 4

File **L5_E4.sas** contains two versions of the previous program. Submit both programs and compare the output and number of observations read. What do you notice about the results?

```
data work.auemps;
   set orion.sales;
   Bonus=Salary*.10;
   if Country='AU' and Bonus>=3000;
run;
```

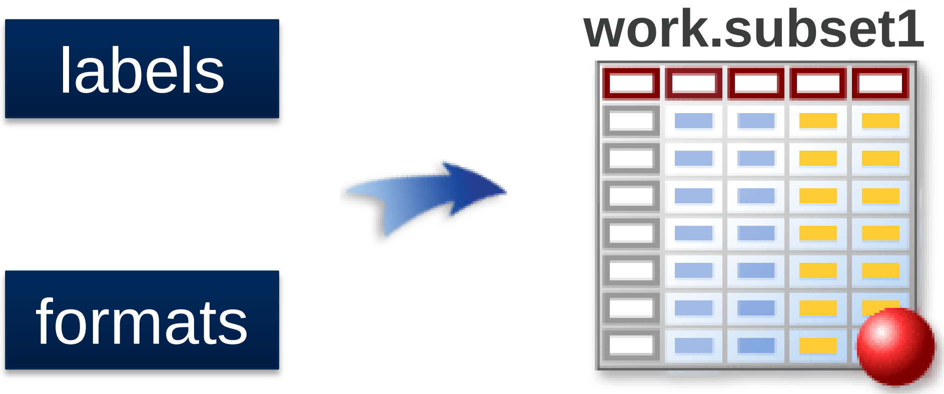


WHERE versus Subsetting IF Statement

Step and Usage	WHERE	IF
PROC step	Yes	No
DATA step (source of variable)		
SET statement	Yes	Yes
assignment statement	No	Yes

Scenario: Part 5

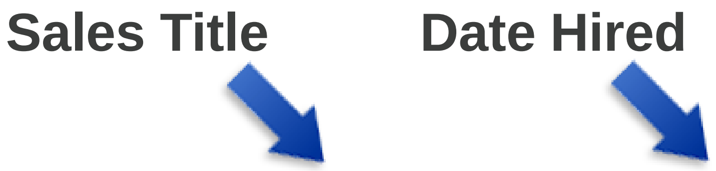
Define permanent labels and formats for some of the variables in the new data set.



LABEL Statement

The LABEL statement assigns descriptive labels to variables.

LABEL statement
DATA Step



work.subset1

First_Name	Last_Name	Salary	Job_Title	Hire_Date	Bonus

Defining Permanent Labels

Use a LABEL statement in a DATA step to permanently assign labels to variables. The labels are stored in the descriptor portion of the data set.

L5_D6.sas

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   label Job_Title='Sales Title'
         Hire_Date='Date Hired';
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

LABEL *variable*='*label* '
      <*variable*='*label* '…>**;**

Viewing the Output

```
proc contents data=work.subset1;
run;
```

Partial PROC CONTENTS Output

L5_D6.sas

```
  Alphabetic List of Variables and Attributes

 #    Variable      Type    Len    Label

 6    Bonus         Num       8
 1    First_Name    Char     12
 5    Hire_Date     Num       8    Date Hired
 4    Job_Title     Char     25    Sales Title
 2    Last_Name     Char     18
 3    Salary        Num       8
```

Viewing the Output: Displaying Labels

To use labels in the PRINT procedure, use the LABEL option in the PROC PRINT statement.

```
proc print data=work.subset1 label;
run;
```

L5_D6.sas

Partial PROC PRINT Output

```
       First_                                                Date
Obs    Name         Last_Name     Salary    Sales Title      Hired       Bonus

  1    Irenie       Elvish         26600    Sales Rep. II     6575      2660.0
  2    Christina    Ngan           27475    Sales Rep. II     8217      2747.5
  3    Kimiko       Hotstone       26190    Sales Rep. I     10866      2619.0
  4    Lucian       Daymond        26480    Sales Rep. I      8460      2648.0
  5    Fong         Hofmeister     32040    Sales Rep. IV     8460      3204.0
```

Viewing the Output: Splitting Labels

Use the PROC PRINT SPLIT= option to split labels across lines based on a split character.

```
proc print data=work.subset1 split=' ';
run;
```

L5_D6.sas

Partial PROC PRINT Output

```
       First_                          Sales                  Date
Obs    Name         Last_Name    Salary    Title                  Hired      Bonus

  1    Irenie       Elvish        26600    Sales Rep. II           6575     2660.0
  2    Christina    Ngan          27475    Sales Rep. II           8217     2747.5
  3    Kimiko       Hotstone      26190    Sales Rep. I           10866     2619.0
  4    Lucian       Daymond       26480    Sales Rep. I            8460     2648.0
  5    Fong         Hofmeister    32040    Sales Rep. IV           8460     3204.0
```

FORMAT Statement

The FORMAT statement associates formats with variables.

FORMAT statement
DATA Step

commax8. ddmmyy10. commax8.2

work.subset1

First_Name	Last_Name	Salary	Job_Title	Hire_Date	Bonus
Irenie	Elvish	26600	Sales Rep. II	6575	2660.0

Defining Permanent Formats

Use a FORMAT statement in a DATA step to permanently associate formats with variables.

L5_D7.sas

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep';
   Bonus=Salary*.10;
   label Job_Title='Sales Title'
         Hire_Date='Date Hired';
   format Salary commax8. Bonus commax8.2
          Hire_Date ddmmyy10.;
   drop Employee_ID Gender Country
        Birth_Date;
run;
```

FORMAT *variable(s) format* …;

Viewing the Output

```
proc contents data=work.subset1;
run;
```

Partial PROC CONTENTS

L5_D7.sas

```
Alphabetic List of Variables and Attributes

#     Variable        Type     Len    Format        Label

6     Bonus           Num        8    COMMAX8.2
1     First_Name      Char      12
5     Hire_Date       Num        8    DDMMYY10.     Date Hired
4     Job_Title       Char      25                  Sales Title
2     Last_Name       Char      18
3     Salary          Num        8    COMMAX8.
```

Viewing the Output

```
proc print data=work.subset1 label;
run;
```

L5_D7.sas

Partial PROC PRINT Output

```
     First_
Obs  Name       Last_Name    Salary  Sales Title     Date Hired       Bonus

  1  Irenie     Elvish       26.600  Sales Rep. II   01/01/1978  2.660,00
  2  Christina  Ngan         27.475  Sales Rep. II   01/07/1982  2.747,50
  3  Kimiko     Hotstone     26.190  Sales Rep. I    01/10/1989  2.619,00
  4  Lucian     Daymond      26.480  Sales Rep. I    01/03/1983  2.648,00
  5  Fong       Hofmeister   32.040  Sales Rep. IV   01/03/1983  3.204,00
```