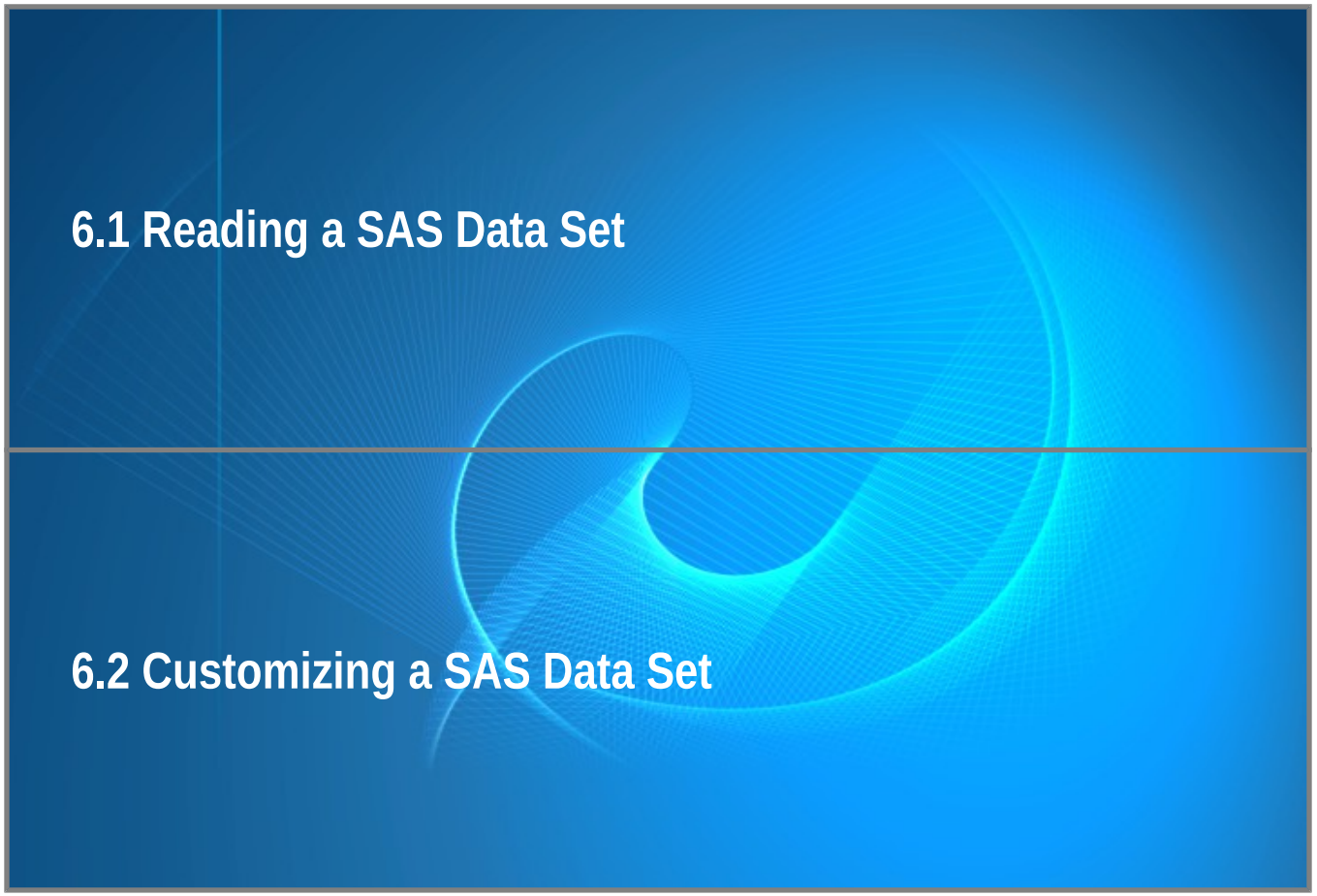
Chapter 6: Reading SAS® Data Sets



Chapter 6: Reading SAS® Data Sets

Objectives

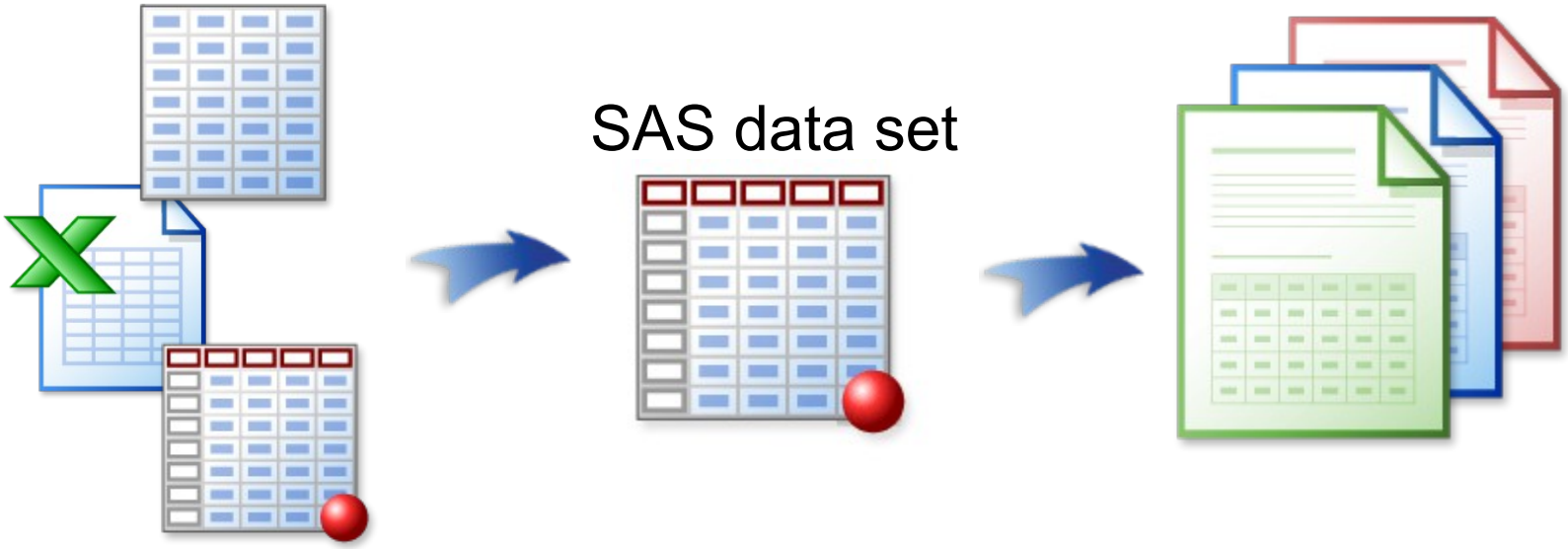
- Define the business scenario that will be used when reading from a data source to create a SAS data set.
- Use a DATA step to create a SAS data set from an existing SAS data set.
- Subset observations with a WHERE statement.
- Create a new variable with an assignment statement.

Business Scenario

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

Considerations

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



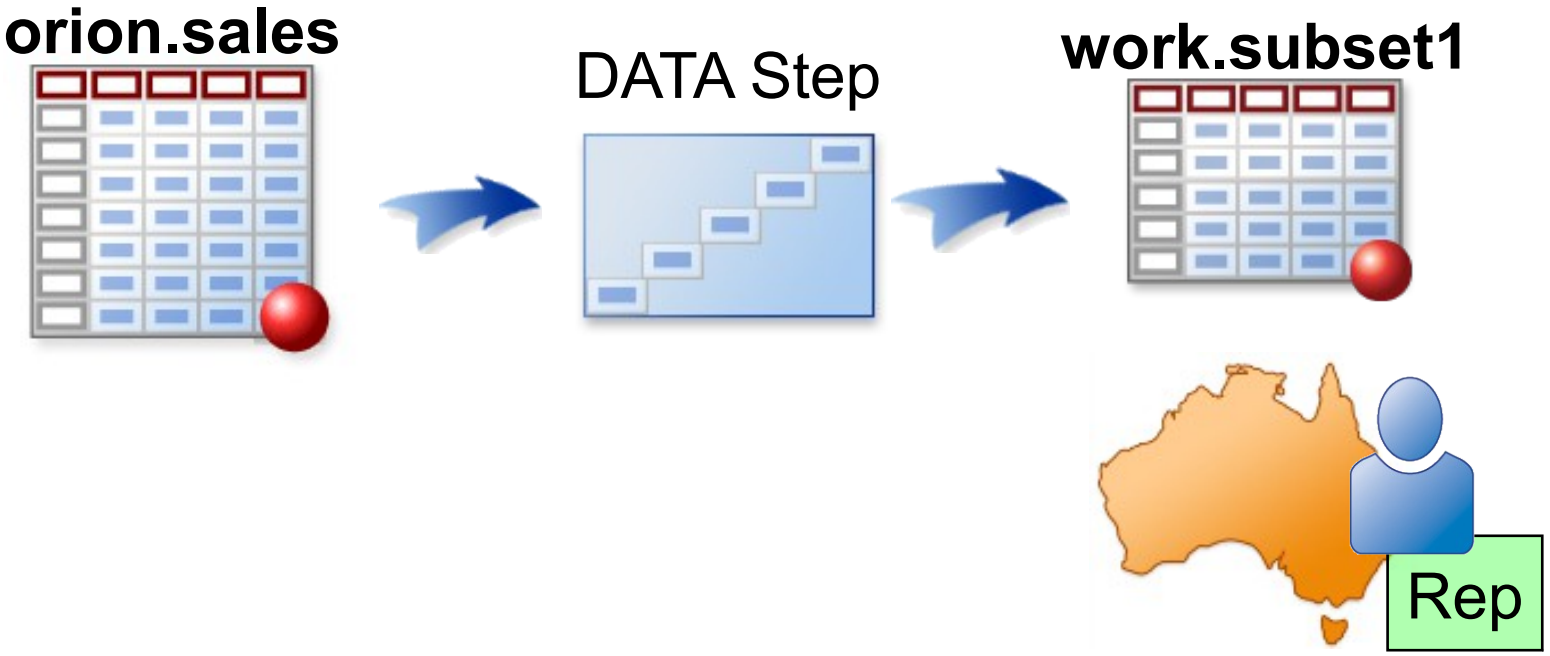
6.01 Multiple Answer Poll

Which types of files will you read into SAS?

- SAS data sets
- Excel worksheets
- database tables
- raw data files
- other
- not sure

Business 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;

p106d01

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.

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

p106d01

SET Statement

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

SET *input-SAS-data-set*;

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

- 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.

p106d01

WHERE Statement

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

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

WHERE *WHERE-expression*;

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

Viewing the Log

Partial SAS 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

Employee_ID	First_ Name	Last_Name	Gender	Salary	Job_Title	Country	Birth_ Date	Hire_ 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

p106d01

Setup for the Poll

Consider the DATA step below.

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

p106a01

6.02 Multiple Choice Poll

Considering this DATA step, which statement is true?

- It reads a temporary data set and creates a permanent data set.
- It reads a permanent data set and creates a temporary data set.
- It contains a syntax error and will not execute.
- It will not execute because you cannot work with permanent and temporary data sets in the same step.

6.02 Multiple Choice Poll – Correct Answer

Considering this DATA step, which statement is true?

- It reads a temporary data set and creates a permanent data set.
- It reads a permanent data set and creates a temporary data set.
- It contains a syntax error and will not execute.
- It will 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;
```

Business 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?

Date Constant

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

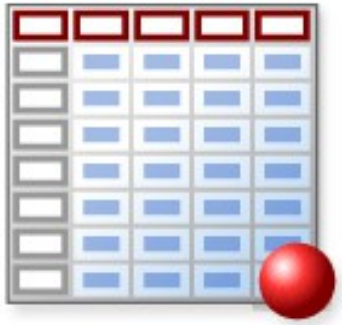
```
data work.subset1;
   set orion.sales;
   where Country='AU' and
         Job_Title contains 'Rep' and
         Hire_Date<'01jan2000'd;
run;
```

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

Considerations

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

orion.sales



work.subset1

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*;**

p106d02a

Assignment Statement

The *expression* consists of operands and operators.

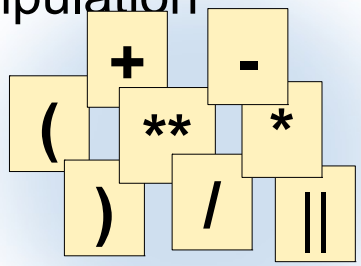
***variable*=*expression*;**

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

Partial SAS 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

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

p106d02a

6.03 Quiz

Evaluate the assignment statements below given the values shown in the PDV.

x	y	z
.	4	10

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

---

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

6.03 Quiz – Correct Answer

Evaluate the assignment statements below given the values shown in the PDV.

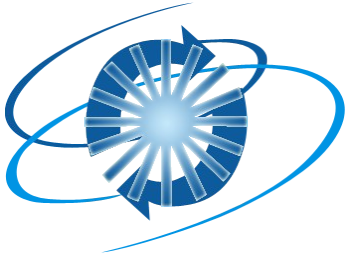
x	y	z
.	4	10

a. `num=y+z/2;` → 4+10/2 → 4+5 → 9

`num=(y+z)/2;` → 14/2 → 7

---

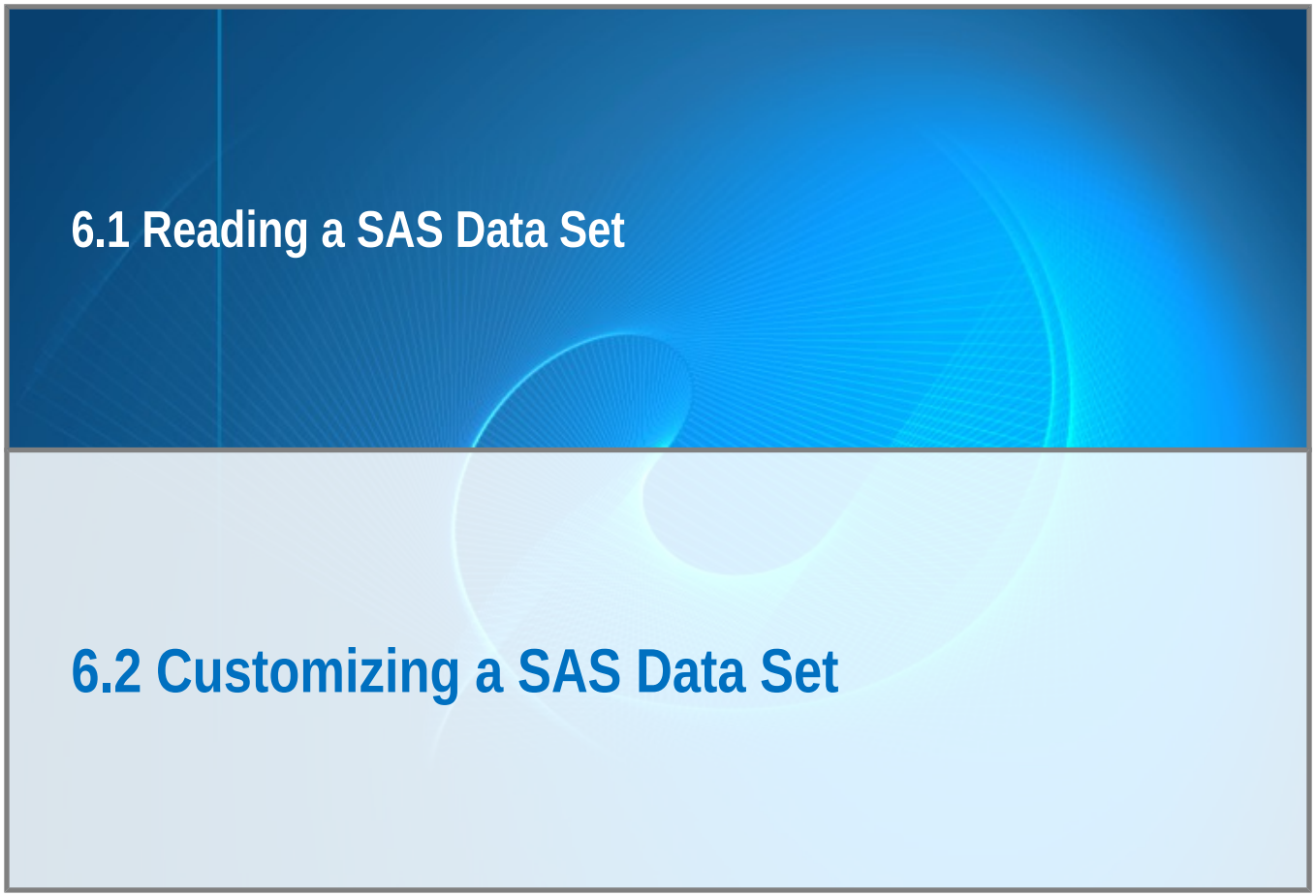
b. `num=x+z/2;` → .+10/2 → .+5 → .



Exercise

This exercise reinforces the concepts discussed previously.

Chapter 6: Reading SAS® Data Sets

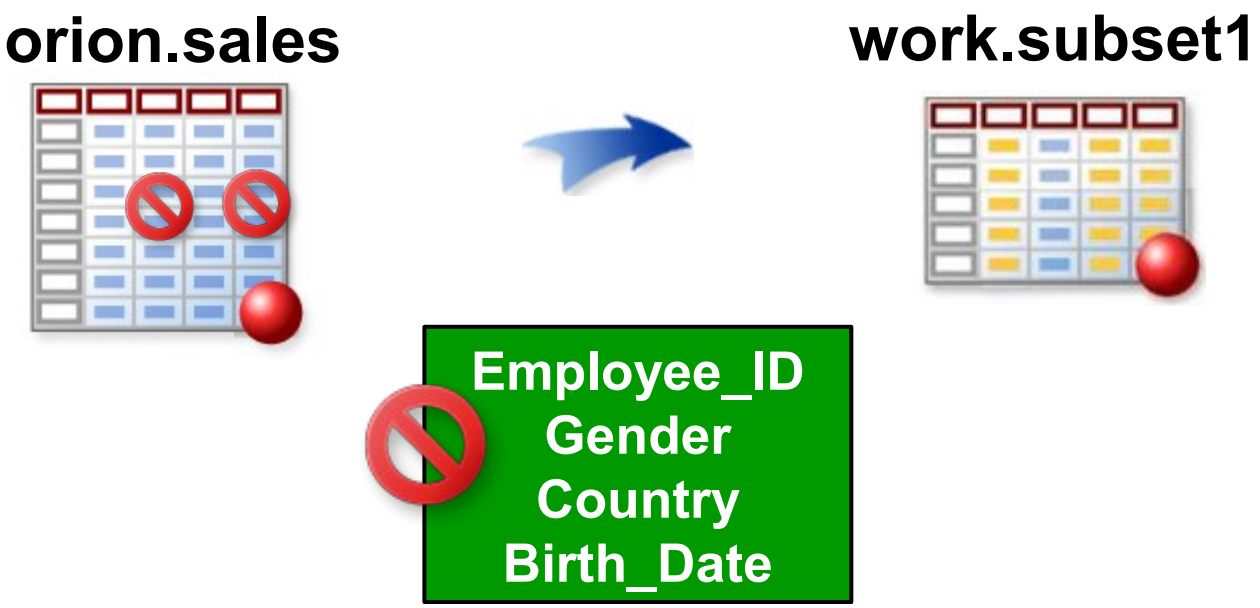


Objectives

- 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.

Business Scenario: Part 3

All Australian sales representatives will get 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.

```
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*;

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

Obs	First_ Name	Last_Name	Salary	Job_Title	Hire_ 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

p106d03

KEEP Statement

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

```
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

Obs	First_ Name	Last_Name	Salary	Job_Title	Hire_ 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

p106d04

Business 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

Execution Phase

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;
```

p106d03

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

Country D $ 2	Birth_Date D 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

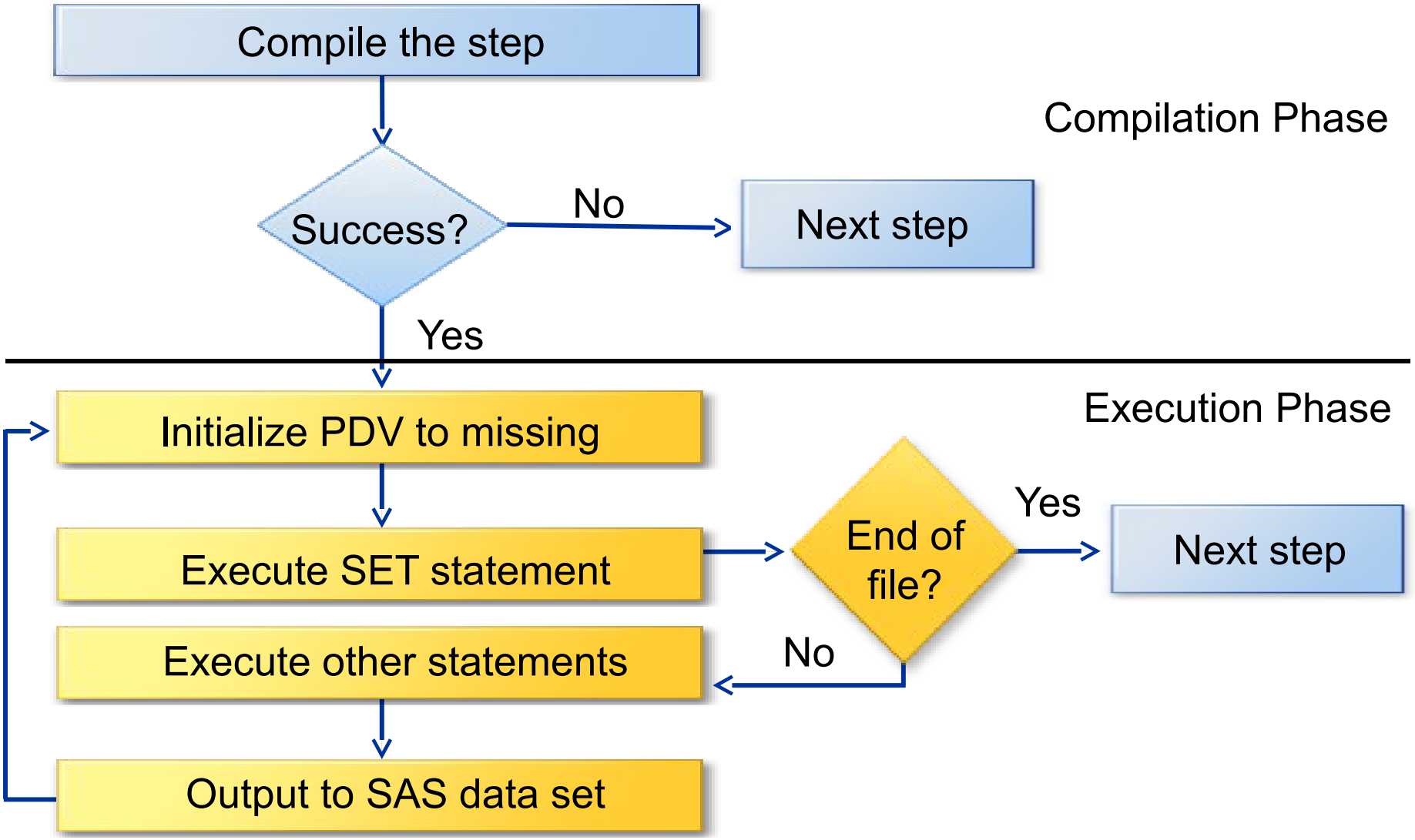
Employee_ID D N 8	First_Name $ 12	Last_Name $ 18	Gender D $ 1	Salary N 8	Job_Title $ 25

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

Descriptor Portion of **work.subset1**

First_Name	Last_Name	Salary	Job_Title	Hire_Date	Bonus

Execution Phase



Execution

Partial **orion.sales**

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

Initialize PDV

```
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	...	Salary	...	Country	Birth_Date	Hire_Date	Bonus
D .		.		D	D .	.	.

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

Partial **orion.sales**

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

Reinitialize PDV

```
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

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
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	...	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	274.75

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	274.75

work.subset1

First_Name	Last_Name	Salary	Job_Title	Hire_Date	Bonus
Irenie	Elvish	26600	Sales Rep II.	6575	2660.00
Christina	Ngan	27475	Sales Rep II.	8217	2747.50

...

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

Employee_ID (D)	...	Salary	...	Country (D)	Birth_Date (D)	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	2660.00
Christina	Ngan	27475	Sales Rep II.	8217	2747.50

Viewing the Output

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

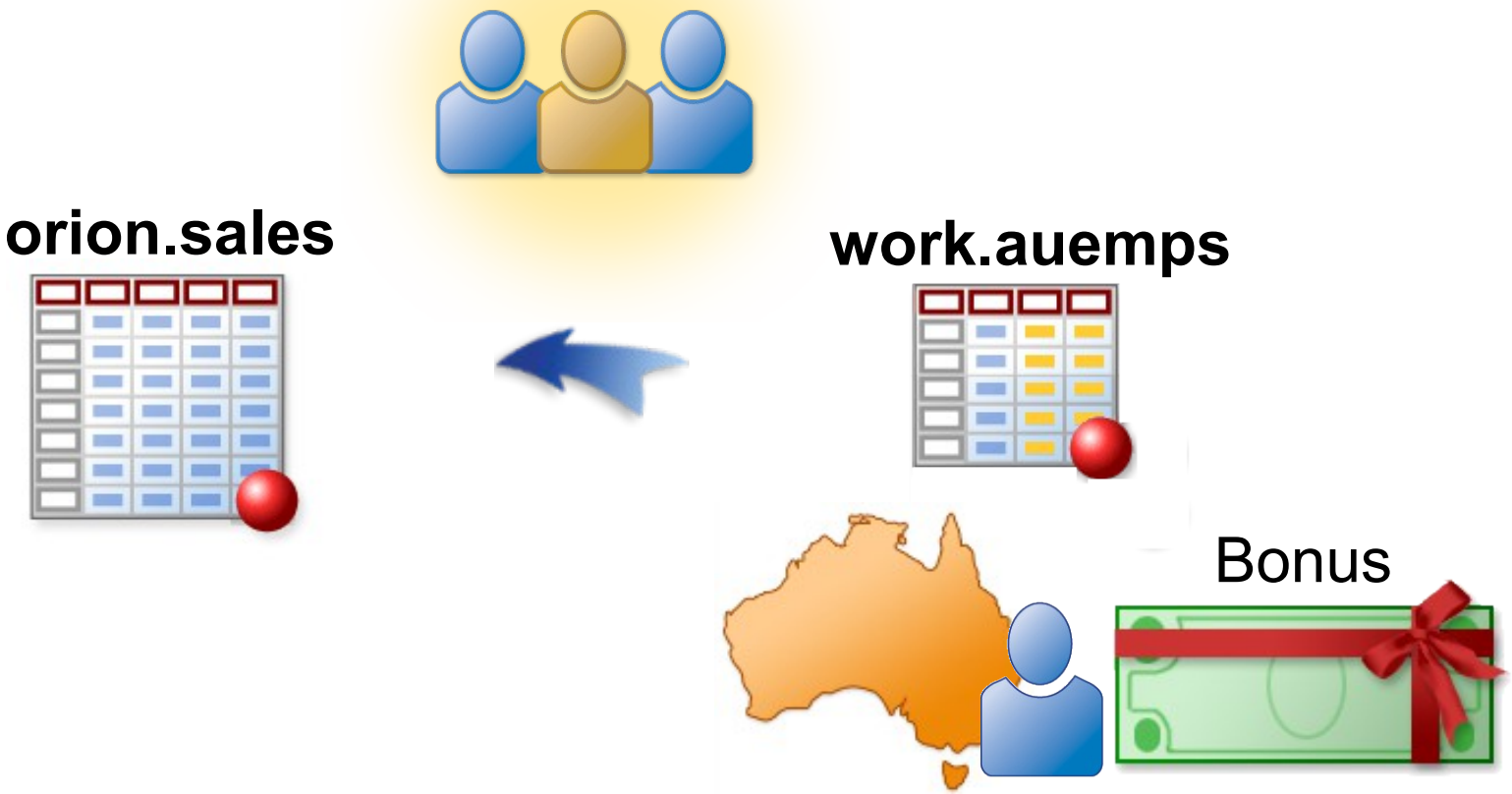
Partial PROC PRINT Output

Obs	First_ Name	Last_Name	Salary	Job_Title	Hire_ 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

p106d03

Business 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.

p106d03

6.04 Quiz

Open and submit **p106a03**. Is the output data set created successfully?

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

p106a03

6.04 Quiz – Correct Answer

Open and submit **p106a03**. Is the output data set created successfully?

```
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 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.

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

IF *condition*;

In this program, processing will reach the bottom of the DATA step and output an observation only if the condition is true.

p106d05

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.

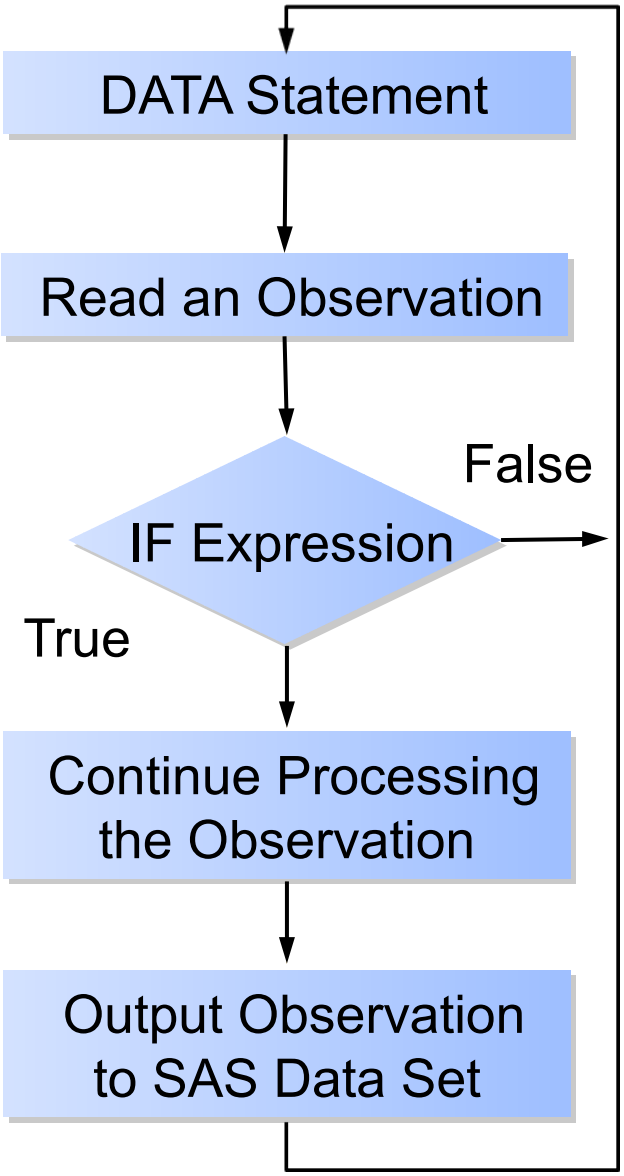
Viewing the Output

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

PROC PRINT Output

Obs	First_ 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



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

Idea Exchange

File **p106a04** 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?

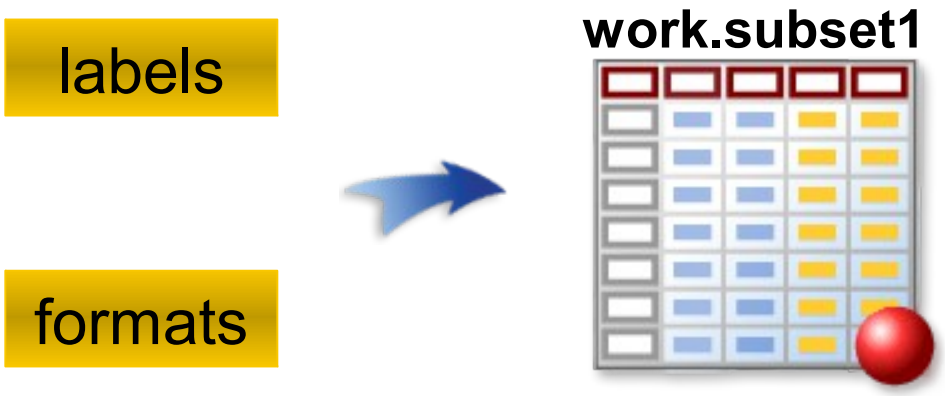
```
data work.auemps;
   set orion.sales;
   Bonus=Salary*.10;
   if Country='AU' and Bonus>=3000;
run;
```

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

Business 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.

Sales Title **Date Hired**

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.

```
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'*…>;

p106d06

Viewing the Output

```
proc contents data=work.subset1;

run;
```

Partial PROC CONTENTS Output

```
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
```


p106d06

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;
```

Partial PROC PRINT Output

Obs	First_ Name	Last_Name	Salary	Sales Title	Date 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

p106d06

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;
```

Partial PROC PRINT Output

Obs	First_ Name	Last_Name	Salary	Sales Title	Date 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

p106d06

6.05 Quiz

What column heading will be displayed for **Job_Title** in the program below?

```
data work.us;
   set orion.sales;
   where Country='US';
   Bonus=Salary*.10;
   label Job_Title='Sales Title';
   drop Employee_ID Gender Country
        Birth_Date;
run;
proc print data=work.subset1 label;
   label Job_Title='Title';
run;
```

p106a05

6.05 Quiz – Correct Answer

What column heading will be displayed for **Job_Title** in the program below?

```
data work.us;
   set orion.sales;
   where Country='US';
   Bonus=Salary*.10;
   label Job_Title='Sales Title';
   drop Employee_ID Gender Country
        Birth_Date;
run;
proc print data=work.subset1 label;
   label Job_Title='Title';
run;
```

The column heading will be Title. Labels and formats in PROC steps override permanent labels and formats.

FORMAT Statement

The FORMAT statement associates formats with variables.

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.

```
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 …*;

p106d07

Viewing the Output

```
proc contents data=work.subset1;

run;
```

Partial PROC CONTENTS Output

```
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

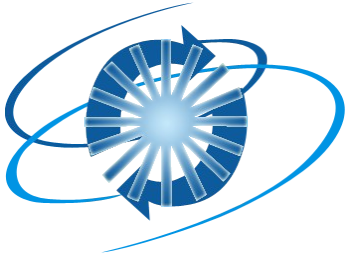
```
proc print data=work.subset1 label;

run;
```

Partial PROC PRINT Output

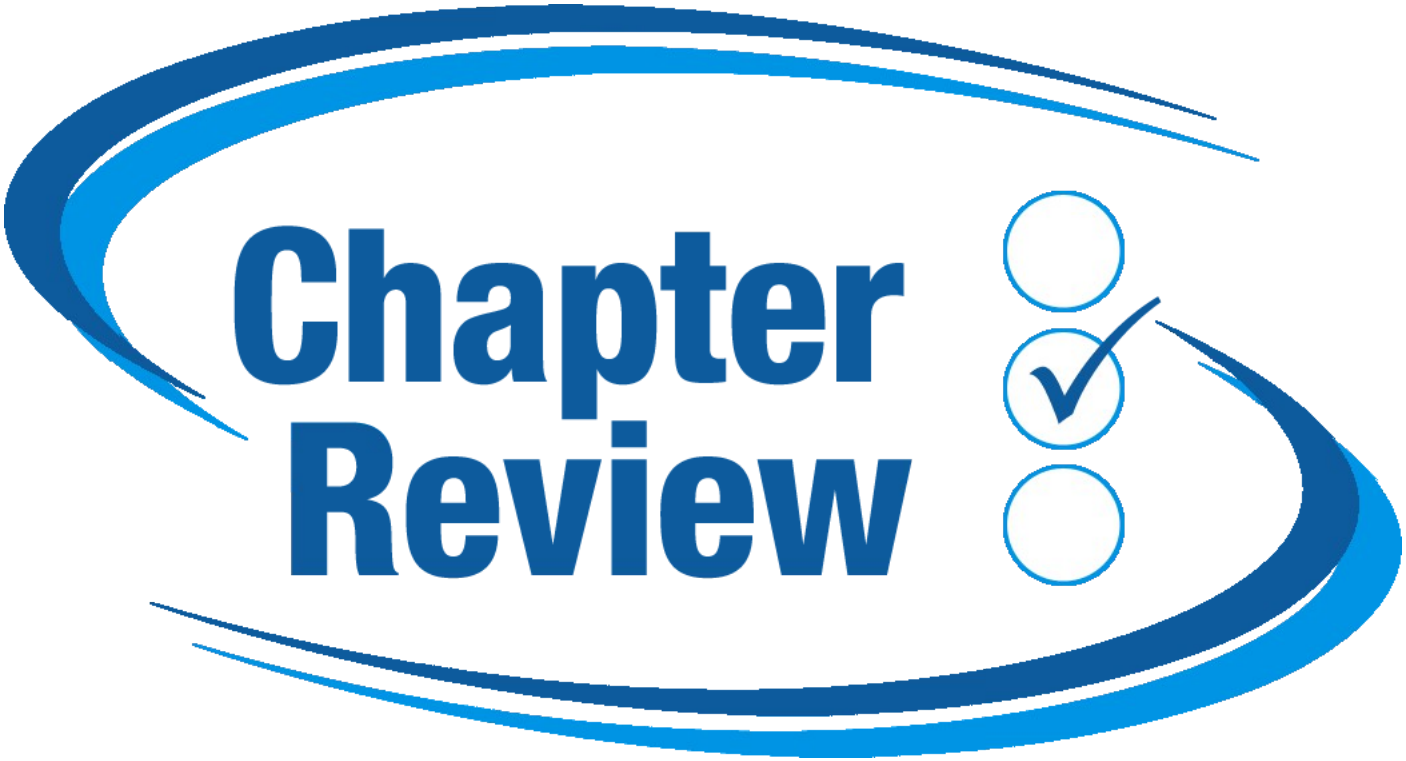
Obs	First_ Name	Last_Name	Salary	Sales Title	Date Hired	Bonus
1	Irenie	Elvish	26.600	Sales Rep. II	01/01/1978	2.660,00
2	Christina	Ngan	27.475	Sales Rep. II	01/07/1982	2.747,50
3	Kimiko	Hotstone	26.190	Sales Rep. I	01/10/1989	2.619,00
4	Lucian	Daymond	26.480	Sales Rep. I	01/03/1983	2.648,00
5	Fong	Hofmeister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

p106d07



Exercise

This exercise reinforces the concepts discussed previously.



1. Which statement is used to read a SAS data set in a DATA step?

   a. DATA statement
   b. WHERE statement
   c. SET statement
   d. assignment statement

1. Which statement is used to read a SAS data set in a DATA step?

   a. DATA statement
   b. WHERE statement
   c. SET statement
   d. assignment statement

2. What is the name of the input data set in the program below?

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

a. work.us
b. orion.sales
c. Country
d. sales

2. What is the name of the input data set in the program below?

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

a. work.us
b. orion.sales
c. Country
d. sales

3. What is the name of the output data set in the program below?

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

- work.us
- orion.sales
- Country
- sales

3. What is the name of the output data set in the program below?

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

- work.us
- orion.sales
- Country
- sales

4. Which of the following DATA steps correctly reads the permanent data set **salesinfo** from the **sporting** library and creates a new data set named **salesinfo2** in the same library?

a.
```
data sporting.salesinfo2;
    set salesinfo;
run;
```
b.
```
data salesinfo2;
    set sporting.salesinfo;
run;
```
c.
```
data sporting.salesinfo2;
    set sporting.salesinfo;
  run;
```

4. Which of the following DATA steps correctly reads the permanent data set **salesinfo** from the **sporting** library and creates a new data set named **salesinfo2** in the same library?

a.
```
data sporting.salesinfo2;
    set salesinfo;
run;
```

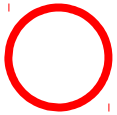
b.
```
data salesinfo2;
    set sporting.salesinfo;
run;
```

c.
```
data sporting.salesinfo2;
    set sporting.salesinfo;
run;
```

5. Which of the following is ***not*** created during the compilation phase?

   - the descriptor portion of the output data set
   - the first observation
   - the program data vector (PDV)

5. Which of the following is ***not*** created during the compilation phase?



- the descriptor portion of the output data set
- the first observation
- the program data vector (PDV)

6. The data set **orion.sales** contains nine variables. Given this DATA step, how many variables does the descriptor portion of **work.comp** contain?

```
data work.comp;
   set orion.sales;
   drop Gender Salary Birth_Date;
run;
```

- 6
- 7
- 9
- None. This program contains a logic error.

6. The data set **orion.sales** contains nine variables. Given this DATA step, how many variables does the descriptor portion of **work.comp** contain?

```
data work.comp;
   set orion.sales;
   drop Gender Salary Birth_Date;
run;
```

- 6
- 7
- 9
- None. This program contains a logic error.

7. The data set **orion.sales** contains nine variables. Given this DATA step, how many variables does the descriptor portion of **work.comp** contain?

```
data work.comp;
   set orion.sales;
   keep Employee_ID Gender Job_Title Salary;
run;
```

- 4
- 9
- 13
- None. This program contains a logic error.

7. The data set **orion.sales** contains nine variables. Given this DATA step, how many variables does the descriptor portion of **work.comp** contain?

```
data work.comp;
   set orion.sales;
   keep Employee_ID Gender Job_Title Salary;
run;
```

- 4
- 9
- 13
- None. This program contains a logic error.

8. A KEEP statement in a DATA step omits all variables except **Name**, **Color**, and **Price** from the output data set. Are the omitted variables included in the PDV?

   - Yes
   - No

8. A KEEP statement in a DATA step omits all variables except **Name**, **Color**, and **Price** from the output data set. Are the omitted variables included in the PDV?

- Yes
- No

9. What value will be assigned to **Units**?

```
data work.comp;
   set work.sales;
   Units=Total+Bonus/Quantity;
run;
```

Partial PDV

Total	Quantity	Bonus	Units
140	10	50	.

- 19
- 145
- 3
- missing

9. What value will be assigned to **Units**?

```
data work.comp;
    set work.sales;
    Units=Total+Bonus/Quantity;
run;
```

Partial PDV

Total	Quantity	Bonus	Units
140	10	50	.

- 19
- 145
- 3
- missing

10. Which procedure can be used to view the permanent labels and formats stored in a data set?

- PROC CONTENTS
- PROC PRINT
- PROC FORMAT
- PROC UNIVARIATE

10. Which procedure can be used to view the permanent labels and formats stored in a data set?

- PROC CONTENTS
- PROC PRINT
- PROC FORMAT
- PROC UNIVARIATE