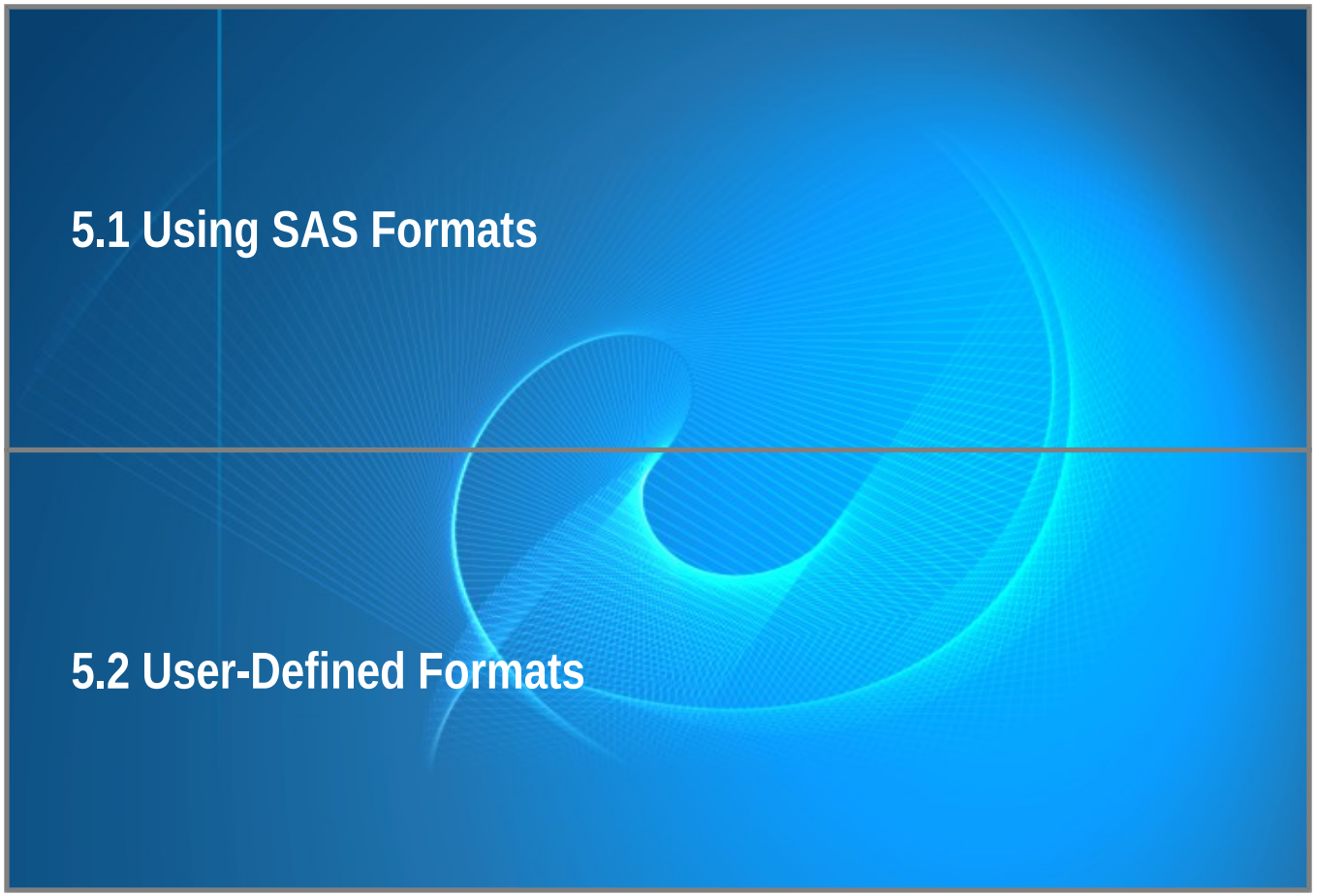
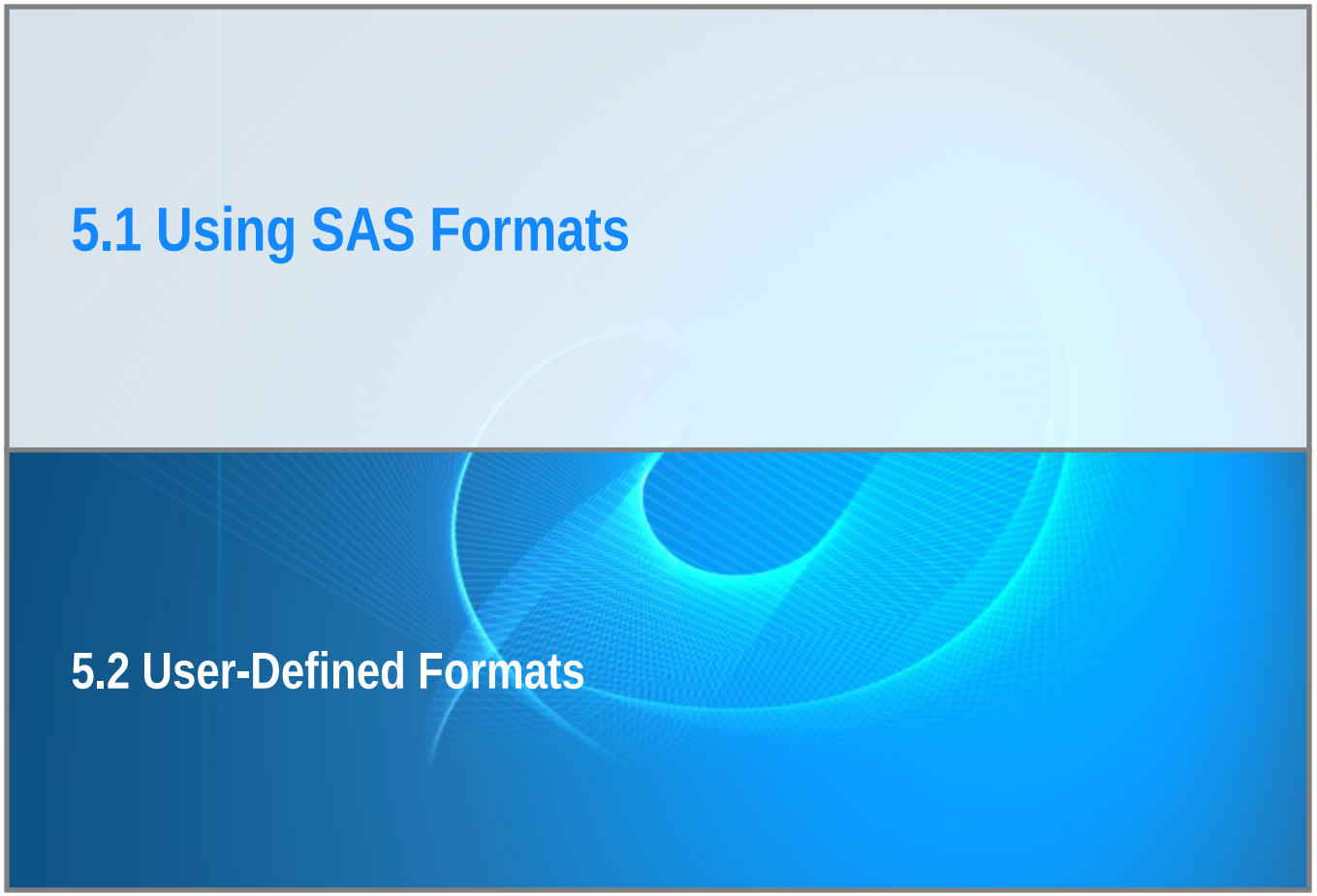
Chapter 5: Formatting Data Values



Chapter 5: Formatting Data Values



Objectives

- Describe SAS formats.
- Apply SAS formats with the FORMAT statement.

Business Scenario

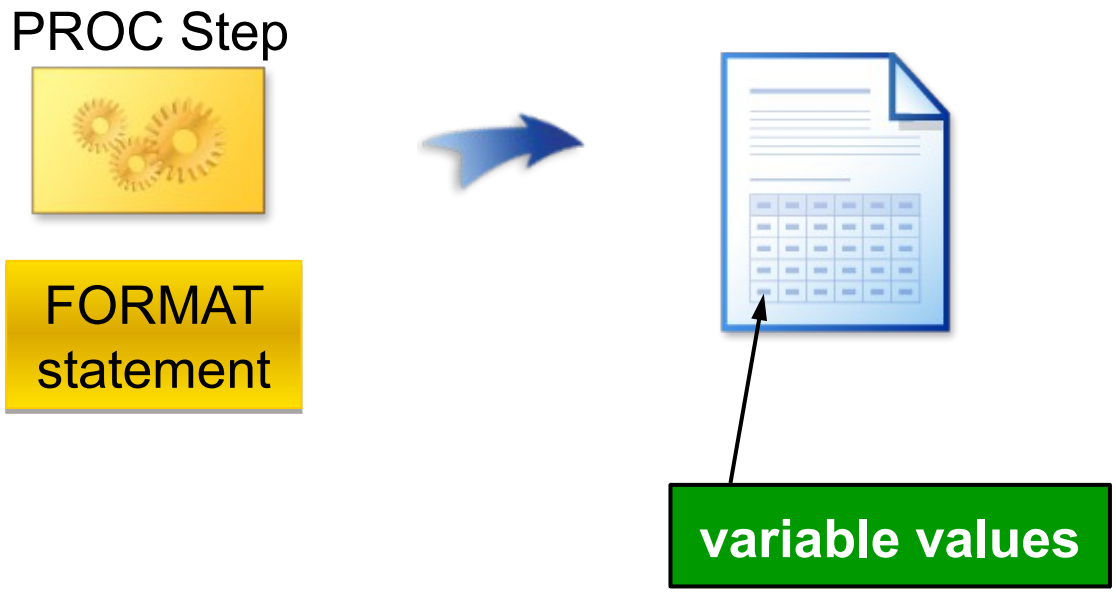
Enhance the appearance of variable values in reports.

Last_Name	First_ Name	Country	Job_Title	Salary	Hire_ Date
Zhou	Tom	AU	Sales Manager	108255	12205
Dawes	Wilson	AU	Sales Manager	87975	6575
Elvish	Irenie	AU	Sales Rep. II	26600	6575

Last_Name	First_ Name	Country	Job_Title	Salary	Hire_Date
Zhou	Tom	AU	Sales Manager	$108,255	06/01/1993
Dawes	Wilson	AU	Sales Manager	$87,975	01/01/1978
Elvish	Irenie	AU	Sales Rep. II	$26,600	01/01/1978

SAS Formats

SAS formats can be used in a PROC step to change how values are displayed in a report.



FORMAT Statement

The *FORMAT statement* associates a format with a variable.

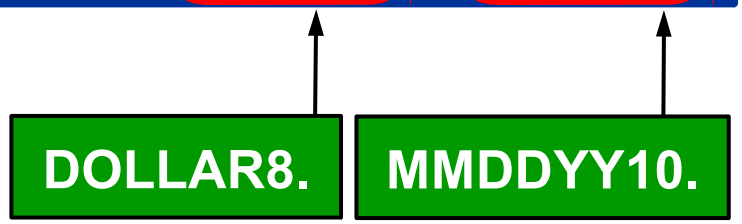
```
proc print data=orion.sales noobs;
    format Salary dollar8. Hire_Date mmddyy10.;
    var Last_Name First_Name Country
        Job_Title Salary Hire_Date;
run;
```

FORMAT *variable(s) format …*;

Viewing the Output

Partial PROC PRINT Output

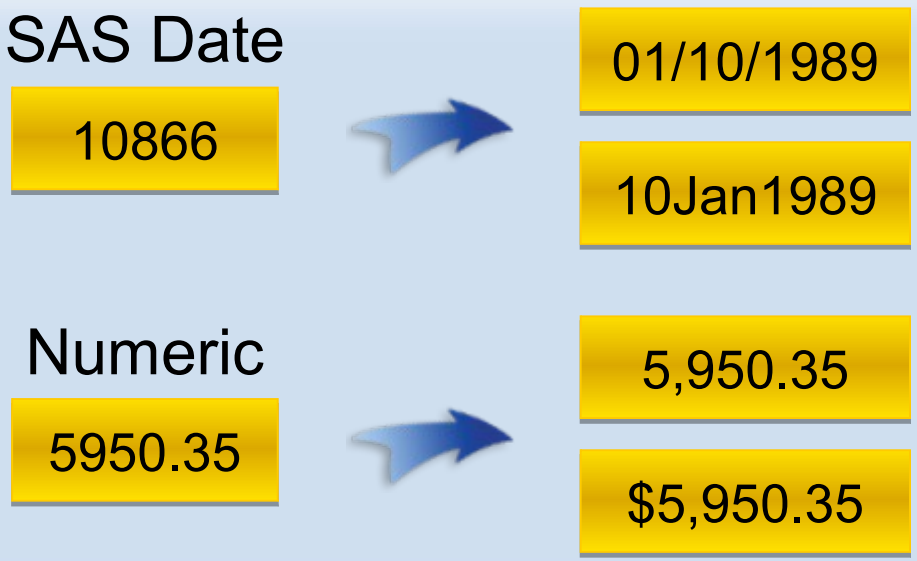
Last_Name	First_ Name	Country	Job_Title	Salary	Hire_Date
Zhou	Tom	AU	Sales Manager	$108,255	06/01/1993
Dawes	Wilson	AU	Sales Manager	$87,975	01/01/1978
Elvish	Irenie	AU	Sales Rep. II	$26,600	01/01/1978
Ngan	Christina	AU	Sales Rep. II	$27,475	07/01/1982
Hotstone	Kimiko	AU	Sales Rep. I	$26,190	10/01/1989



What Is a Format?

A *format* is an instruction to write data values.

- A format changes the appearance of a variable's value in a report.
- The values stored in the data set are ***not*** changed.



SAS Formats

SAS formats have the following form:

<$>*format*<*w*>.<*d*>

$	Indicates a character format.
format	Names the SAS format.
w	Specifies the total format width, including decimal places and special characters.
.	Is required syntax. Formats always contain a period (.) as part of the name.
d	Specifies the number of decimal places to display in numeric formats.

SAS Formats

Selected SAS formats:

Format	Definition
$*w*.	Writes standard character data.
w.*d*	Writes standard numeric data.
COMMA*w*.*d*	Writes numeric values with a comma that separates every three digits and a period that separates the decimal fraction.
DOLLAR*w*.*d*	Writes numeric values with a leading dollar sign, a comma that separates every three digits, and a period that separates the decimal fraction.
COMMAX*w*.*d*	Writes numeric values with a period that separates every three digits and a comma that separates the decimal fraction.
EUROX*w*.*d*	Writes numeric values with a leading euro symbol (€), a period that separates every three digits, and a comma that separates the decimal fraction.

SAS Format Examples

Selected SAS formats:

Format	Stored Value	Displayed Value
$4.	Programming	Prog
12.	27134.5864	27135
12.2	27134.5864	27134.59
COMMA12.2	27134.5864	27,134.59
DOLLAR12.2	27134.5864	$27,134.59
COMMAX12.2	27134.5864	27.134,59
EUROX12.2	27134.5864	€27.134,59

SAS Format Examples

If the format width is not large enough to accommodate a numeric value, the displayed value is automatically adjusted to fit the width.

Format	Stored Value	Displayed Value
DOLLAR12.2	27134.5864	$27,134.59
DOLLAR9.2	27134.5864	$27134.59
DOLLAR8.2	27134.5864	27134.59
DOLLAR5.2	27134.5864	27135
DOLLAR4.2	27134.5864	27E3

5.01 Quiz

Use SAS documentation or the SAS Help Facility to explore the *Zw.d* numeric format. What is it used for?

Hint: Search for *Zw.d* or explore “Formats by Category.”

5.01 Quiz – Correct Answer

Use SAS documentation or the SAS Help Facility to explore the *Zw.d* numeric format. What is it used for?

Hint: Search for *Zw.d* or explore "Formats by Category."

The *Zw.d* format writes standard numeric data with leading zeros. It is similar to the *w.d* format except that *Zw.d* pads right-aligned output with zeros instead of blanks.

SAS Date Format Examples

SAS date formats display SAS date values in standard date forms.

Format	Stored Value	Displayed Value
MMDDYY10.	0	01/01/1960
MMDDYY8.	0	01/01/60
MMDDYY6.	0	010160
DDMMYY10.	365	31/12/1960
DDMMYY8.	365	31/12/60
DDMMYY6.	365	311260

SAS Date Format Examples

Additional date formats:

Format	Stored Value	Displayed Value
DATE7.	-1	31DEC59
DATE9.	-1	31DEC1959
WORDDATE.	0	January 1, 1960
WEEKDATE.	0	Friday, January 1, 1960
MONYY7.	0	JAN1960
YEAR4.	0	1960

5.02 Quiz

Which FORMAT statement creates the output shown below?

```
format Birth_Date Hire_Date mmddyy10.
       Term_Date monyy7.;
```

```
format Birth_Date Hire_Date ddmmyyyy.
       Term_Date mmmyyyy.;
```

```
format Birth_Date Hire_Date ddmmyy10.
       Term_Date monyy7.;
```

- Answer

output

Birth_Date	Hire_Date	Term_Date
21/05/1969	15/10/1992	MAR2007

5.02 Quiz – Correct Answer

Which FORMAT statement creates the output shown below?

```
format Birth_Date Hire_Date mmddyy10.
       Term_Date monyy7.;
```

```
format Birth_Date Hire_Date ddmmyyyy.
       Term_Date mmmyyyy.;
```

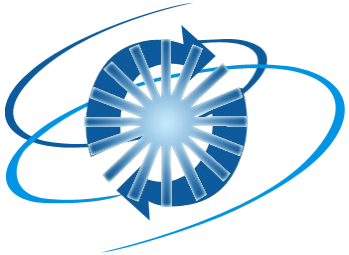
○

```
format Birth_Date Hire_Date ddmmyy10.
       Term_Date monyy7.;
```

- Answer

output

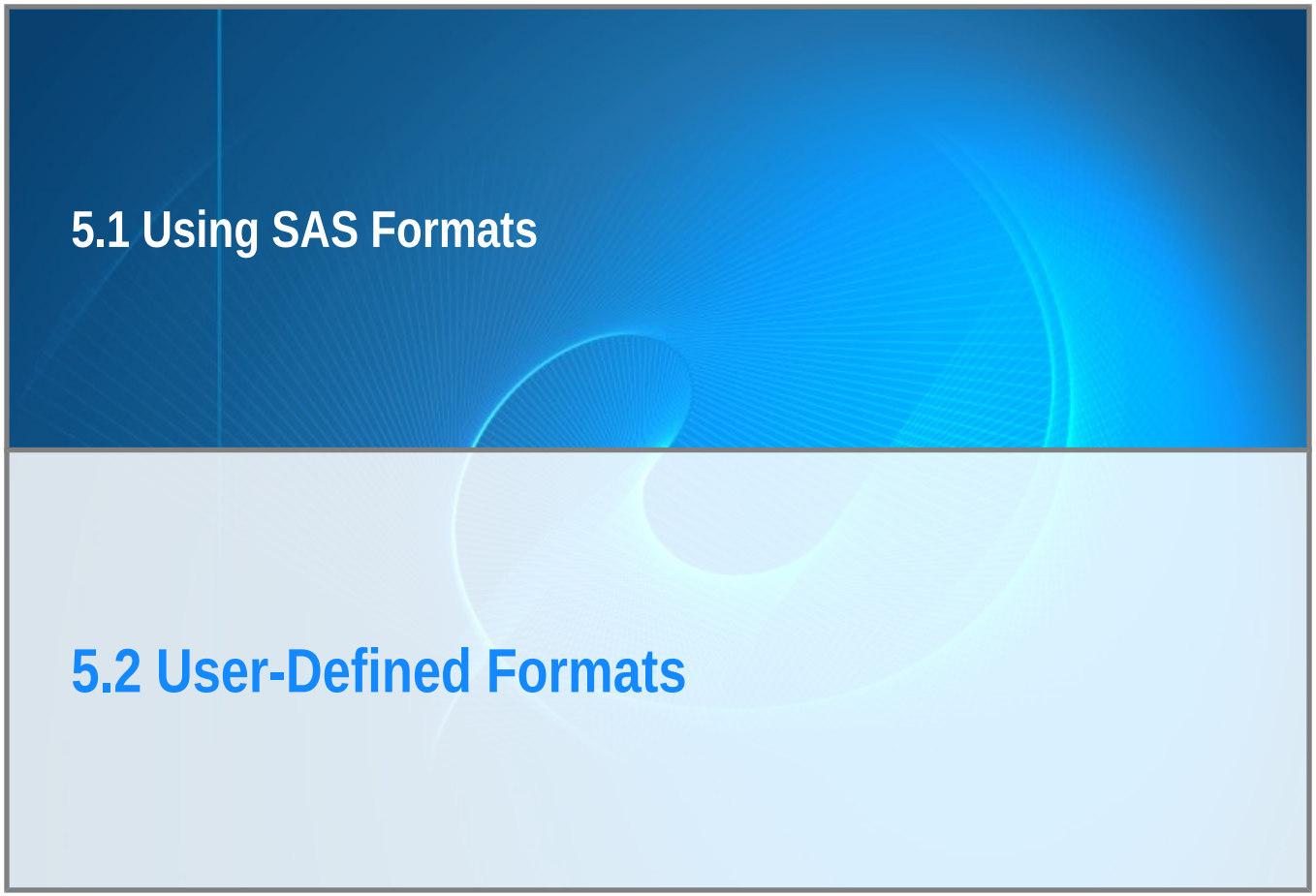
Birth_Date	Hire_Date	Term_Date
21/05/1969	15/10/1992	MAR2007



Exercise

This exercise reinforces the concepts discussed previously.

Chapter 5: Formatting Data Values



Objectives

- Create user-defined formats using the FORMAT procedure.
- Apply user-defined formats using a FORMAT statement in a report.
- Use formats to recode data values.
- Use formats to collapse or aggregate data.

Business Scenario

Display country names instead of country codes in a report.

Current Report (partial output)

Obs	Employee_ID	Salary	Country	Birth_ Date	Hire_ Date
1	120102	$108,255	AU	AUG1973	JUN1993
2	120103	$87,975	AU	JAN1953	JAN1978
3	120121	$26,600	AU	AUG1948	JAN1978

Desired Report (partial output)

Obs	Employee_ID	Salary	Country	Birth_ Date	Hire_ Date
1	120102	$108,255	Australia	AUG1973	JUN1993
2	120103	$87,975	Australia	JAN1953	JAN1978
3	120121	$26,600	Australia	AUG1948	JAN1978

p105d02

User-Defined Formats: Part 1

Use PROC FORMAT to create a user-defined format.

```
proc format;
   value $ctryfmt   'AU'='Australia'
                    'US'='United States'
                   other='Miscoded';
run;
```

PROC FORMAT;
VALUE *format-name range1* **=** *'label'*
range2 **=** *'label'*
. . . **;**
RUN;

p105d03

User-Defined Formats: Part 2

Use a FORMAT statement in the PROC PRINT step to apply the format to a specific variable.

```
proc print data=orion.sales;
   var Employee_ID Job_Title Salary
       Country Birth_Date Hire_Date;
   format Salary dollar10.
          Birth_Date Hire_Date monyy7.
          Country $ctryfmt.;
run;
```

p105d03

Viewing the Output

Partial PROC PRINT Output

Obs	Employee_ID	Salary	Country	Birth_ Date	Hire_ Date
1	120102	$108,255	Australia	AUG1973	JUN1993
2	120103	$87,975	Australia	JAN1953	JAN1978
3	120121	$26,600	Australia	AUG1948	JAN1978
4	120122	$27,475	Australia	JUL1958	JUL1982
5	120123	$26,190	Australia	SEP1968	OCT1989

VALUE Statement

```
VALUE format-name range1='label '
                  range2='label '
                  . . . ;
```

A format name

- can be up to 32 characters in length
- for character formats, must begin with a dollar sign ($), followed by a letter or underscore
- for numeric formats, must begin with a letter or underscore
- cannot end in a number
- cannot be given the name of a SAS format
- cannot include a period in the VALUE statement.

VALUE Statement

```
VALUE format-name range1='label '
                  range2='label '
                  . . . ;
```

Each range can be

- a single value
- a range of values
- a list of values.

Labels

- can be up to 32,767 characters in length
- are enclosed in quotation marks.

5.03 Multiple Answer Poll

Which names are invalid for user-defined formats?

- $stfmt
- $3levels
- _4years
- salranges
- dollar

5.03 Multiple Answer Poll – Correct Answer

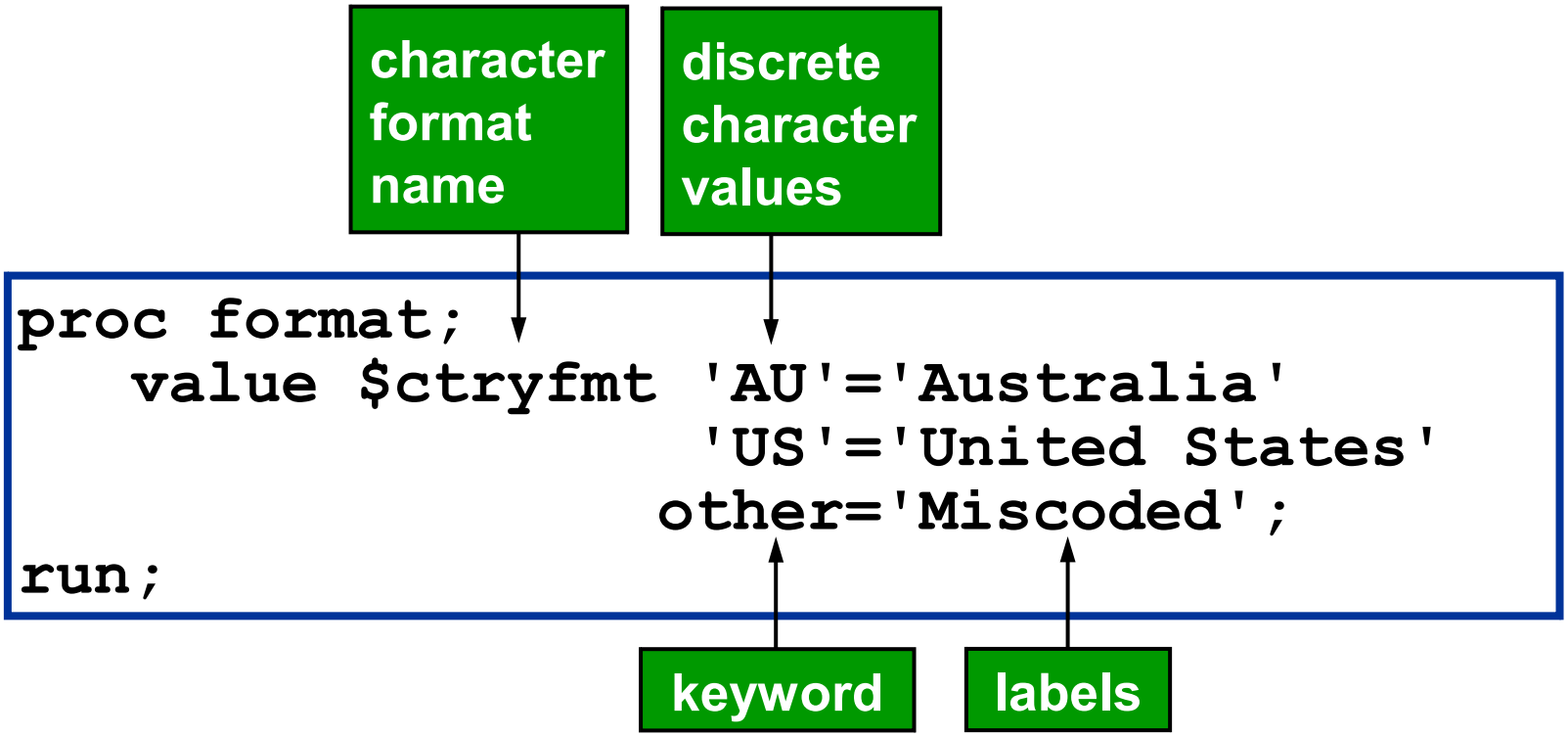
Which names are invalid for user-defined formats?

- $stfmt
- $3levels
- _4years
- salranges
- dollar

Character formats must have a dollar sign as the first character and a letter or underscore as the second character.

User-defined formats cannot be given the name of a format provided by SAS.

Defining a Character Format



The OTHER keyword includes all values that do not match any other value or range.

p105d03

Applying a Format

User-defined and SAS formats can be applied in a single FORMAT statement.

```
proc print data=orion.sales label;
   var Employee_ID Job_Title Salary
       Country Birth_Date Hire_Date;
   format Salary dollar10.
          Birth_Date Hire_Date monyy7.
          Country $ctryfmt.;
run;
```

- A period (for example, at the end of the $CTRYFMT format) is required when user-defined formats are used in a FORMAT statement.

p105d03

Idea Exchange

The formatting examples shown in this section are sometimes referred to as *translating values*.

Can you give an example of where this type of application might be useful?

Business Scenario

An Orion Star manager wants a report showing employee salaries collapsed into three user-defined groups or tiers.

Current Report

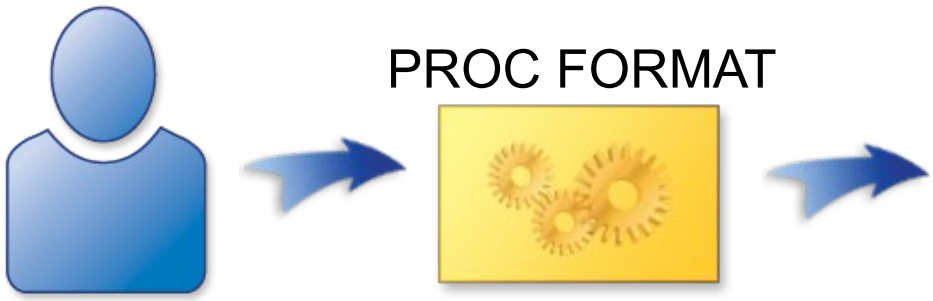
Obs	Employee_ID	Last_Name	Salary
1	120102	Zhou	108255
2	120103	Dawes	87975
3	120121	Elvish	26600
4	120122	Ngan	27475

ired Report

Obs	Employee_ID	Last_Name	Salary
1	120102	Zhou	Tier 3
2	120103	Dawes	Tier 2
3	120121	Elvish	Tier 1
4	120122	Ngan	Tier 1

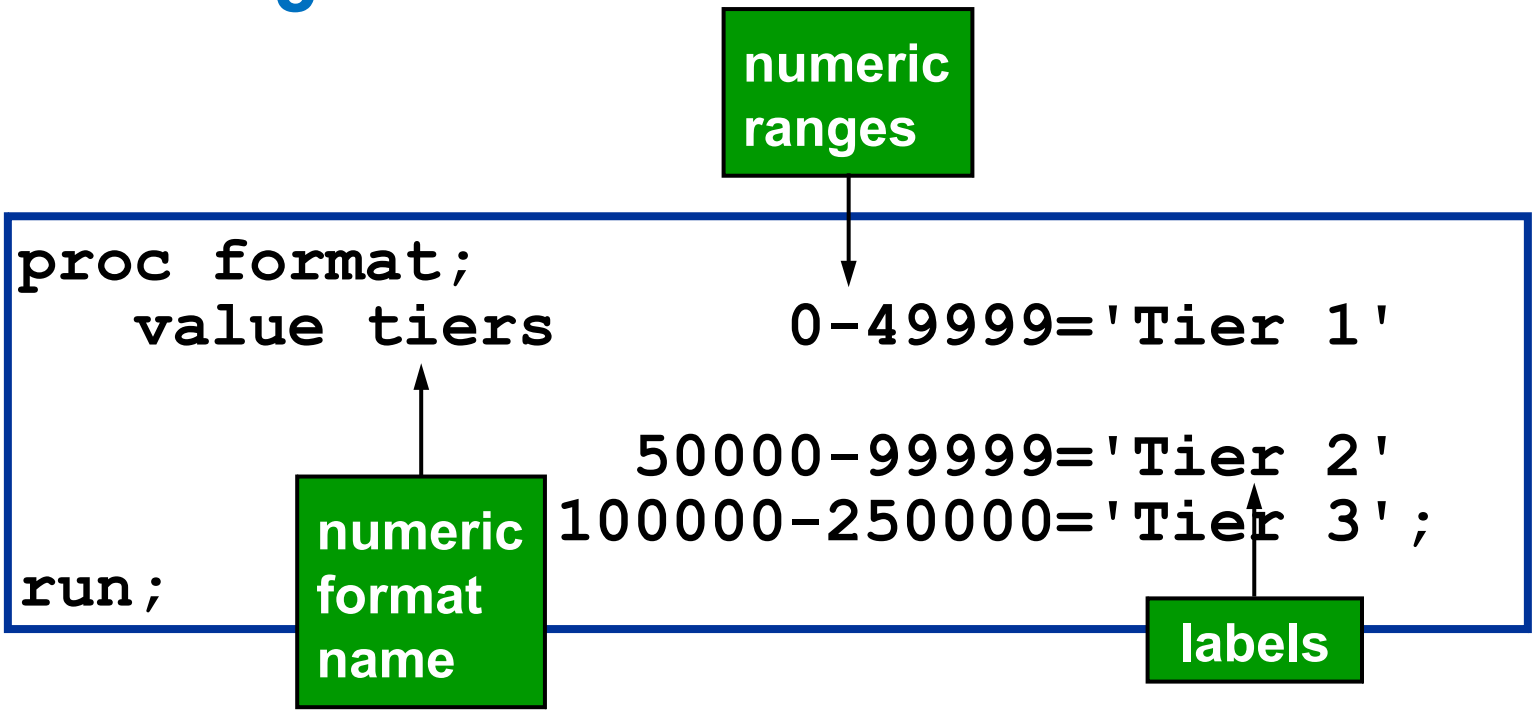
Specifying Ranges of Values

Use PROC FORMAT to specify the salary range for each tier.



Salary	Value
20,000 to 49,999	Tier1
50,000 to 99,999	Tier2
100,000 to 250,000	Tier3

Defining a Numeric Format



p105d04

Defining and Using a Numeric Format

This demonstration illustrates the use of a user-defined numeric format.

p105d04

Defining a Continuous Range

The less than (<) symbol excludes the endpoint from a range, allowing a continuous range.

- Put < after the starting value in a range to exclude it.
- Put < before the ending value in a range to exclude

Range	Starting Value	Ending Value
50000 - 100000	Includes 50000	Includes 100000
50000 - < 100000	Includes 50000	Excludes 100000
50000 < - 100000	Excludes 50000	Includes 100000
50000 < - < 100000	Excludes 50000	Excludes 100000

5.04 Quiz

How will a value of *50000* be displayed if the TIERS format below is applied to the value?

- Tier 1
- Tier 2
- 50000
- a missing value

```
proc format;
   value tiers  20000-<50000 ='Tier 1'

                50000-<100000='Tier 2'
                100000-250000='Tier 3';
run;
```

p105d05

5.04 Quiz – Correct Answer

How will a value of *50000* be displayed if the TIERS format below is applied to the value?

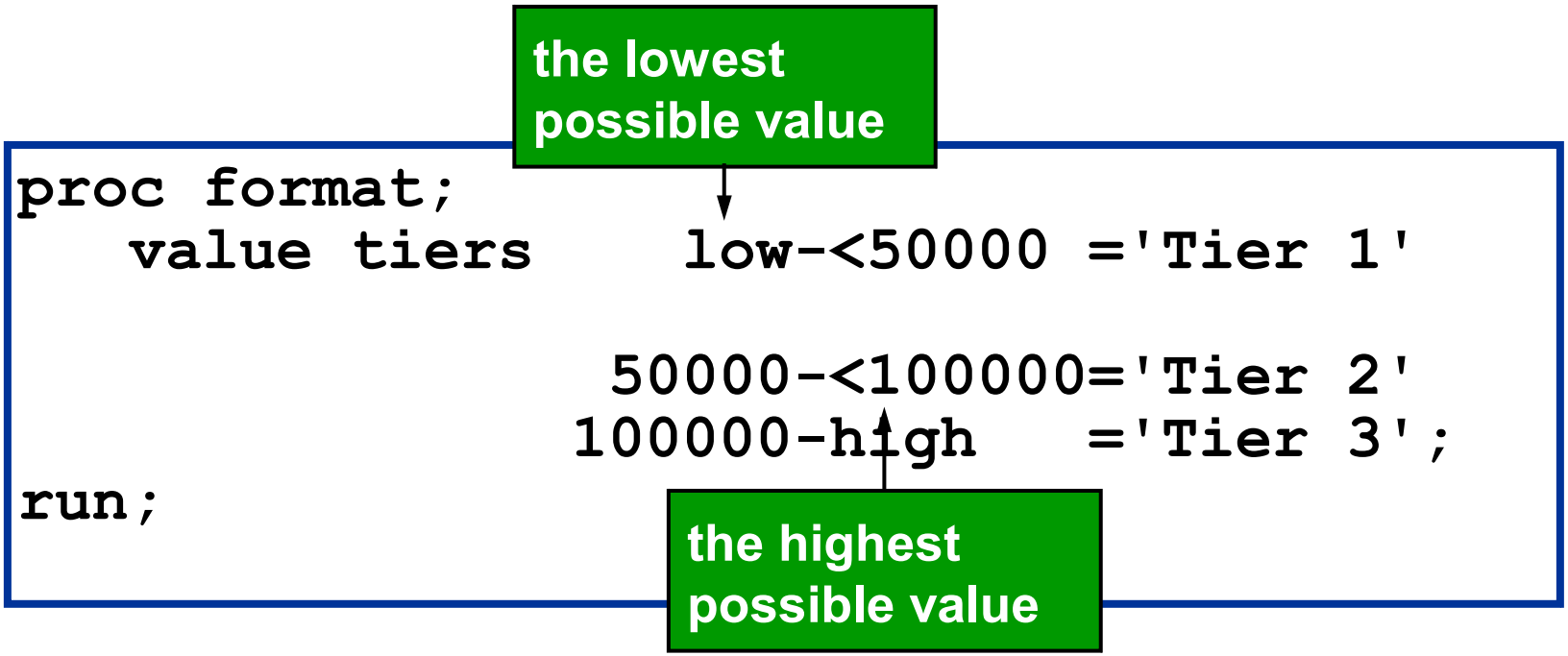
- Tier 1
- Tier 2
- 50000
- a missing value

```
proc format;
   value tiers  20000-<50000 ='Tier 1'

                50000-<100000='Tier 2'
                100000-250000='Tier 3';
run;
```

p105d05

LOW and HIGH Keywords



The LOW keyword

- includes missing values for character variables
- does not include missing values for numeric variables.

p105d06

Applying a Numeric Format

Part 1

```
proc format;
   value tiers     low-<50000 ='Tier 1'

                   50000-<100000='Tier 2'
                   100000-high  ='Tier 3';
run;
```

Part 2

```
proc print data=orion.sales;
   var Employee_ID Job_Title Salary
       Country Birth_Date Hire_Date;
   format Birth_Date Hire_Date monyy7.
          Salary tiers.;
run;
```


p105d06

Viewing the Output

Partial PROC PRINT Output

Obs	Employee_ID	Job_Title	Salary	Birth_ Country	Hire_ Date	Date
1	120102	Sales Manager	Tier 3	AU	AUG1973	JUN1993
2	120103	Sales Manager	Tier 2	AU	JAN1953	JAN1978
3	120121	Sales Rep. II	Tier 1	AU	AUG1948	JAN1978
4	120122	Sales Rep. II	Tier 1	AU	JUL1958	JUL1982
5	120123	Sales Rep. I	Tier 1	AU	SEP1968	OCT1989

User-Defined Format Example

Ranges can be specified using lists, ranges, discrete values, and keywords.

```
proc format;
   value mnthfmt 1,2,3='Qtr 1'
                 4-6='Qtr 2'
                 7-9='Qtr 3'
                10-12='Qtr 4'
                   .='missing'
                other='unknown';
run;
```

Multiple User-Defined Formats

Multiple VALUE statements can be included in a single PROC FORMAT step.

```
proc format;
   value $ctryfmt   'AU'='Australia'
                    'US'='United States'
                   other='Miscoded';

   value tiers      low-<50000 ='Tier 1'

                  50000-<100000='Tier 2'
                  100000-high  ='Tier 3';
run;
```

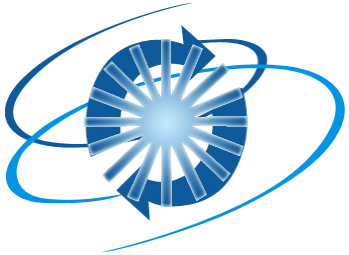
Viewing the Output

```
proc print data=orion.sales;
   var Employee_ID Job_Title Salary
       Country Birth_Date Hire_Date;
   format Birth_Date Hire_Date monyy7.
          Country $ctryfmt.
          Salary tiers.;
run;
```

Partial PROC PRINT Output

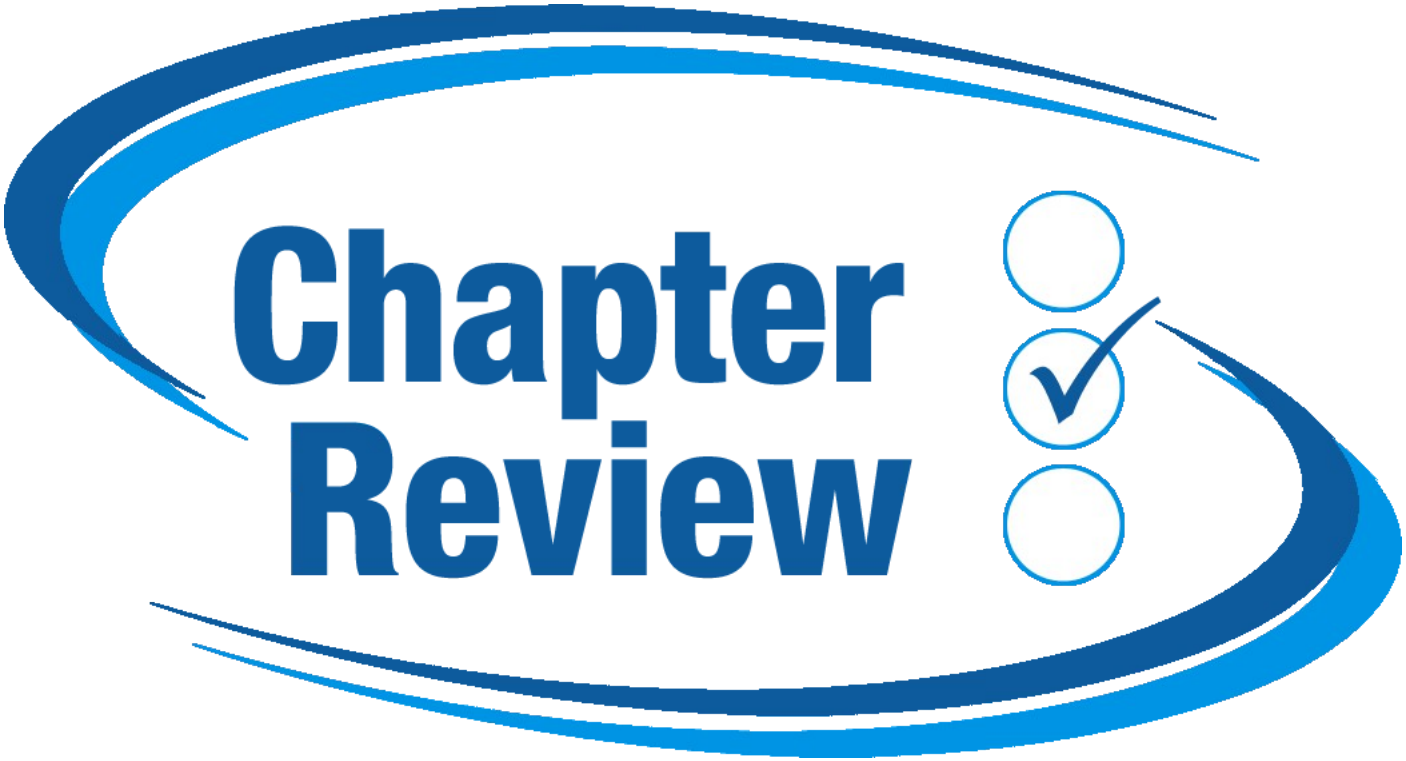
Obs	Employee_ID	Job_Title	Salary	Country	Birth_ Date	Hire_ Date
1	120102	Sales Manager	Tier 3	Australia	AUG1973	JUN1993
2	120103	Sales Manager	Tier 2	Australia	JAN1953	JAN1978
3	120121	Sales Rep. II	Tier 1	Australia	AUG1948	JAN1978
4	120122	Sales Rep. II	Tier 1	Australia	JUL1958	JUL1982
5	120123	Sales Rep. I	Tier 1	Australia	SEP1968	OCT1989

p105d07



Exercise

This exercise reinforces the concepts discussed previously.



1. Which of the following is a valid name for a character format?
   - country
   - $ctry
   - $country.
   - _country

1. Which of the following is a valid name for a character format?
   - country
   - $ctry
   - $country.
   - _country

2. You specify the variable to which a format applies when creating it in a PROC FORMAT step.

   - True
   - False

2. You specify the variable to which a format applies when creating it in a PROC FORMAT step.

   - True
   - False

3. Which of the following FORMAT statements was used to create this output?

Employee_ID	Job_Title	Salary
120102	Sales Manager	$108255.00
120103	Sales Manager	$87,975.00
120121	Sales Rep. II	$26,600.00
120122	Sales Rep. II	$27,475.00
120123	Sales Rep. I	$26,190.00

- format Salary dollar.;
- format Salary dollar12.2;
- format Salary dollar11.2;
- format Salary dollar10.2;

3. Which of the following FORMAT statements was used to create this output?

Employee_ID	Job_Title	Salary
120102	Sales Manager	$108255.00
120103	Sales Manager	$87,975.00
120121	Sales Rep. II	$26,600.00
120122	Sales Rep. II	$27,475.00
120123	Sales Rep. I	$26,190.00

- format Salary dollar.;
- format Salary dollar12.2;
- format Salary dollar11.2;
- format Salary dollar10.2;

4. Which of the following FORMAT statements was used to create this output?

Obs	Order_ID	Order_ Date	Delivery_ Date
1	1230058123	11JAN07	01/11/07
2	1230080101	15JAN07	01/19/07
3	1230106883	20JAN07	01/22/07
4	1230147441	28JAN07	01/28/07
5	1230315085	27FEB07	02/27/07

- format Order_Date date9. Delivery_Date mmddyy8.;
- format Order_Date date7. Delivery_Date mmddyy8.;
- format Order_Date ddmmmyy. Delivery_Date mmddyy8.;
- format Order_Date monyy7. Delivery_Date mmddyy8.;

4. Which of the following FORMAT statements was used to create this output?

Obs	Order_ID	Order_ Date	Delivery_ Date
1	1230058123	11JAN07	01/11/07
2	1230080101	15JAN07	01/19/07
3	1230106883	20JAN07	01/22/07
4	1230147441	28JAN07	01/28/07
5	1230315085	27FEB07	02/27/07

- format Order_Date date9. Delivery_Date mmddyy8.;
- format Order_Date date7. Delivery_Date mmddyy8.;
- format Order_Date ddmmmyy. Delivery_Date mmddyy8.;
- format Order_Date monyy7. Delivery_Date mmddyy8.;

5. Which of the following is not a valid user-defined format name?

   - $month
   - group_a
   - comma
   - _gender

5. Which of the following is not a valid user-defined format name?

   - $month
   - group_a
   - comma
   - _gender

6. You can use either < or > to define a non-inclusive range in a VALUE statement.

   - True
   - False

6. You can use either < or > to define a non-inclusive range in a VALUE statement.

   - True
   - False

7. The format name must include a period in the FORMAT statement.

   - True
   - False

7. The format name must include a period in the FORMAT statement.

- True
- False

8. Given this $TITLE format, what would be displayed for a value of *Sales Rep II*?

```
proc format;
   value $title
         'Sales Manager',
         'Senior Sales Mgr'='Manager'
         'Sales Rep. I',
         'Sales Rep. II'='Rep';
run;
```

- Sales Manager
- Rep
- Sales Rep II
- Sales R

8. Given this $TITLE format, what would be displayed for a value of *Sales Rep II*?

```
proc format;
   value $title
         'Sales Manager',
         'Senior Sales Mgr'='Manager'
         'Sales Rep. I',
         'Sales Rep. II'='Rep';
run;
```

- Sales Manager
- Rep
- Sales Rep II
- Sales R

9. A format modifies both the stored value and the displayed value.

   - True
   - False

9. A format modifies both the stored value and the displayed value.

- True
- False

10. A FORMAT statement is used only to apply SAS formats.

- True
- False

10. A FORMAT statement is used only to apply SAS formats.

- True
- False