Practices for Lesson 3: Declaring PL/SQL Variables

Chapter 3

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Practice 3: Declaring PL/SQL Variables

In this practice, you declare PL/SQL variables.

1. Identify valid and invalid identifiers:
   a. `today`
   b. `last_name`
   c. `today's_date`
   d. `Number_of_days_in_February_this_year`
   e. `Isleap$year`
   f. `#number`
   g. `NUMBER#`
   h. `number1to7`
2. Identify valid and invalid variable declaration and initialization:
   a. `number_of_copies PLS_INTEGER;`
   b. `PRINTER_NAME constant VARCHAR2(10);`
   c. `deliver_to VARCHAR2(10)`:=Johnson;
   d. `by_when DATE:= CURRENT_DATE`+1;
3. Examine the following anonymous block, and then select a statement from the following that is true.

```
DECLARE
  v_fname VARCHAR2(20);
  v_lname VARCHAR2(15) DEFAULT 'fernandez';
BEGIN
  DBMS_OUTPUT.PUT_LINE(v_fname ||' ' ||v_lname);
END;
```

   a. The block executes successfully and prints “fernandez.”
   b. The block produces an error because the `fname` variable is used without initializing.
   c. The block executes successfully and prints “null fernandez.”
   d. The block produces an error because you cannot use the `DEFAULT` keyword to initialize a variable of type `VARCHAR2`.
   e. The block produces an error because the `v_fname` variable is not declared.

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

4. Modify an existing anonymous block and save it as a new script.
   a. Open the `lab_02_02_soln.sql` script, which you created in Practice 2 titled "Introduction to PL/SQL."
   b. In this PL/SQL block, declare the following variables:
      1) `v_today` of type `DATE`. Initialize `today` with `SYSDATE`.
      2) `v_tomorrow` of type `today`. Use the `%TYPE` attribute to declare this variable.
   c. In the executable section:
      1) Initialize the `v_tomorrow` variable with an expression, which calculates tomorrow's date (add one to the value in `today`)
      2) Print the value of `v_today` and `v_tomorrow` after printing "Hello World"
   d. Save your script as `lab_03_04_soln.sql`, and then execute.

      The sample output is as follows (the values of `v_today` and `v_tomorrow` will be different to reflect your current today's and tomorrow's date):

```
PL/SQL procedure successfully completed.

 Hello World
TODAY IS : 18-JUL-16
TOMORROW IS : 19-JUL-16
```

5. Edit the `lab_03_04_soln.sql` script.
   a. Add code to create two bind variables named `b_basic_percent` and `b_pf_percent`. Both bind variables are of type `NUMBER`.
   b. In the executable section of the PL/SQL block, assign the values `45` and `12` to `b_basic_percent` and `b_pf_percent`, respectively.
   c. Terminate the PL/SQL block with "`/`" and display the value of the bind variables by using the `PRINT` command.
   d. Execute and save your script as `lab_03_05_soln.sql`. The sample output is as follows:

```
Script Output
Task completed in 0.014 seconds
PL/SQL procedure successfully completed.

Hello World
TODAY IS : 18-JUL-16
TOMORROW IS : 19-JUL-16


B_BASIC_PERCENT
---------------
45

B_PF_PERCENT
------------
12
```

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

Solution 3: Declaring PL/SQL Variables

1. Identify valid and invalid identifiers:
   a. `today` **Valid**
   b. `last_name` **Valid**
   c. `today's_date` **Invalid** – character "'" not allowed
   d. `Number_of_days_in_February_this_year` **Invalid** – Too long
   e. `Isleap$year` **Valid**
   f. `#number` **Invalid –** Cannot start with "#"
   g. `NUMBER#` **Valid**
   h. `number1to7` **Valid**
2. Identify valid and invalid variable declaration and initialization:
   a. `number_of_copies` `PLS_INTEGER;` **Valid**
   b. `PRINTER_NAME` `constant VARCHAR2(10);` **Invalid**
   c. `deliver_to` `VARCHAR2(10):=`Johnson; **Invalid**
   d. `by_when` `DATE:= CURRENT_DATE`+1; **Valid**

*The declaration in **b** is invalid because constant variables must be initialized during declaration.*

*The declaration in **c** is invalid because string literals should be enclosed within single quotation marks.*

3. Examine the following anonymous block, and then select a statement from the following that is true.

```
DECLARE
  v_fname VARCHAR2(20);
  v_lname VARCHAR2(15) DEFAULT 'fernandez';
BEGIN
  DBMS_OUTPUT.PUT_LINE(v_fname ||' ' ||v_lname);
END;
```

   a. The block executes successfully and prints "fernandez."
   b. The block produces an error because the `fname` variable is used without initializing.
   c. The block executes successfully and prints "null fernandez."
   d. The block produces an error because you cannot use the `DEFAULT` keyword to initialize a variable of type `VARCHAR2`.
   e. The block produces an error because the `v_fname` variable is not declared.

   a. The block will execute successfully and print "fernandez."

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

4. Modify an existing anonymous block and save it as a new script.
   a. Open the `lab_02_02_soln.sql` script, which you created in Practice 2 titled "Introduction to PL/SQL."
   b. In the PL/SQL block, declare the following variables:
      1) Variable `v_today` of type `DATE`. Initialize `today` with `SYSDATE`.

```
DECLARE
   v_today DATE:=SYSDATE;
```

      2) Variable `v_tomorrow` of type `today`. Use the `%TYPE` attribute to declare this variable.

```
   v_tomorrow v_today%TYPE;
```

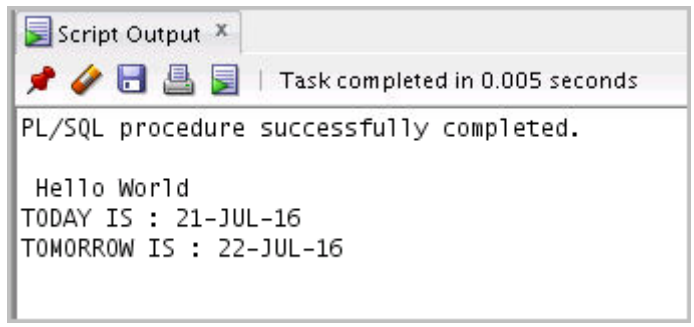
In the executable section:

1) Initialize the `v_tomorrow` variable with an expression, which calculates tomorrow's date (add one to the value in `v_today`)
2) Print the value of `v_today` and `v_tomorrow` after printing "Hello World"

```
BEGIN
   v_tomorrow:=v_today +1;
   DBMS_OUTPUT.PUT_LINE(' Hello World ');
   DBMS_OUTPUT.PUT_LINE('TODAY IS : '|| v_today);
   DBMS_OUTPUT.PUT_LINE('TOMORROW IS : ' || v_tomorrow);
END;
```

   c. Save your script as `lab_03_04_soln.sql`, and then execute.

      The sample output is as follows (the values of `v_today` and `v_tomorrow` will be different to reflect your current today's and tomorrow's date):



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

5. Edit the `lab_03_04_soln.sql` script.
   a. Add code to create two bind variables named `b_basic_percent` and `b_pf_percent`. Both bind variables are of type `NUMBER`.

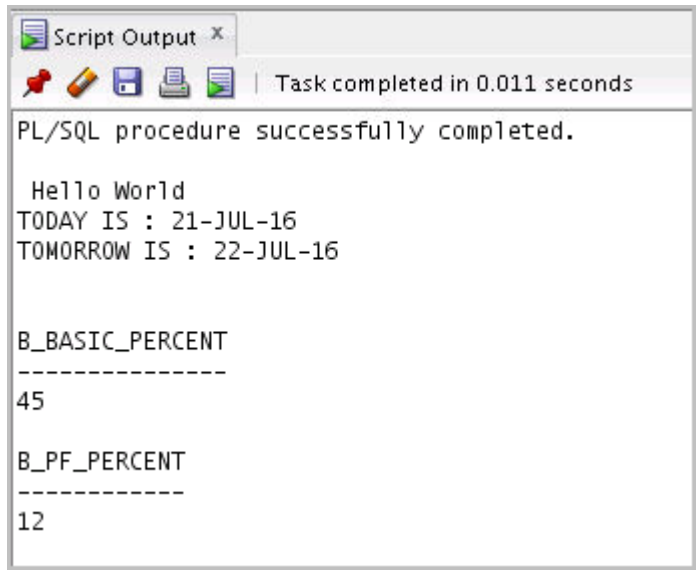
```
VARIABLE b_basic_percent NUMBER
VARIABLE b_pf_percent NUMBER
```

   b. In the executable section of the PL/SQL block, assign the values `45` and `12` to `b_basic_percent` and `b_pf_percent`, respectively.

```
    :b_basic_percent:=45;
    :b_pf_percent:=12;
```

   c. Terminate the PL/SQL block with "/" and display the value of the bind variables by using the `PRINT` command.

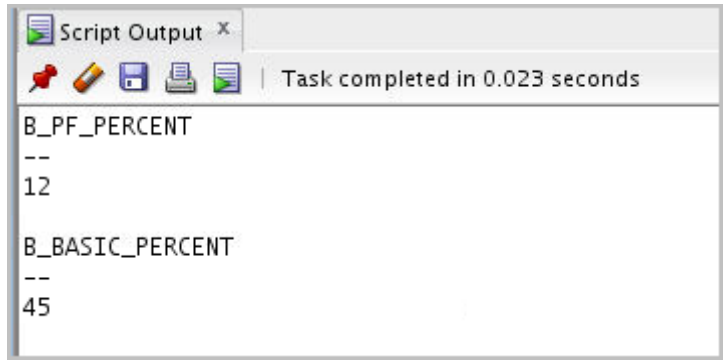
```
/
PRINT b_basic_percent
PRINT b_pf_percent
```



OR

```
PRINT
```

   d. Execute and save your script as `lab_03_05_soln.sql`. The sample output is as follows:



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.