**Quiz**

You can use most SQL single-row functions such as number, character, conversion, and date single-row functions in PL/SQL expressions.

1. True

2. False

**Answer: True**

**Practice 3: Writing Executable Statements**

**In this practice, you examine and write executable statements.**

Изображение выглядит как текст

Автоматически созданное описание

1) Evaluate the preceding PL/SQL block and determine the data type and value of each of the following variables, according to the rules of scoping.

a) The value of v\_weight at position 1 is: **2**

The data type of v\_weight at position 1 is: **NUMBER**

b) The value of v\_new\_locn at position 1 is: **Western Europe**

The data type of v\_new\_locn at position 1 is: **VARCHAR2**

c) The value of v\_weight at position 2 is: **601**

The data type of v\_weight at position 2 is: **NUMBER**

d) The value of v\_message at position 2 is: **Product 10012 is in stock**

The data type of v\_message at position 2 is: **VARCHAR2**

e) The value and data type of v\_new\_locn at position 2 is: **Not defined**

Изображение выглядит как текст

Автоматически созданное описание

2) In the preceding PL/SQL block, determine the values and data types for each of the following cases:

a) The value of v\_customer in the nested block is: **201**

The data type of v\_customer in the nested block is: **NUMBER**

b) The value of v\_name in the nested block is: **Unisports**

The data type of v\_name in the nested block is: **VARCHAR2**

c) The value of v\_credit\_rating in the nested block is: **GOOD**

The data type of v\_credit\_rating in the nested block is: **VARCHAR2**

d) The value of v\_customer in the main block is: **Womansport**

The data type of v\_credit\_rating in the nested block is: **VARCHAR2**

e) The value of v\_name in the main block is: **Not defined**

f) The value of v\_credit\_rating in the main block is: **EXCELLENT**

The data type of v\_credit\_rating in the nested block is: **VARCHAR2**

3) Use the same session that you used to execute the practices in the lesson titled "Declaring PL/SQL variables." If you have opened a new session, execute lab\_02\_05\_soln.sql. Then, edit lab\_02\_05\_soln.sql as follows:

a) Use single-line comment syntax to comment the lines that create the bind variables, and turn on SERVEROUTPUT.

**SET SERVEROUTPUT ON**

**--VARIABLE b\_basic\_percent NUMBER;**

**--VARIABLE b\_pf\_percent NUMBER;**

Изображение выглядит как текст

Автоматически созданное описание

b) Use multiple-line comments in the executable section to comment the lines that assign values to the bind variables.

**/\***

**:b\_basic\_percent := 45;**

**:b\_pf\_percent := 12;**

**\*/**

Изображение выглядит как текст

Автоматически созданное описание

c) In the declaration section:

1. Declare and initialize two temporary variables to replace the commented out bind variables

**DECLARE**

**v\_today DATE := SYSDATE;**

**v\_tomorrow v\_today%TYPE;**

**v\_basic\_percent NUMBER := 45;**

**v\_pf\_percent NUMBER := 12;**

**Изображение выглядит как текст

Автоматически созданное описание**

2. Declare two additional variables: v\_fname of type VARCHAR2 and size 15, and v\_emp\_sal of type NUMBER and size 10

**DECLARE**

**v\_today DATE := SYSDATE;**

**v\_tomorrow v\_today%TYPE;**

**v\_basic\_percent NUMBER := 45;**

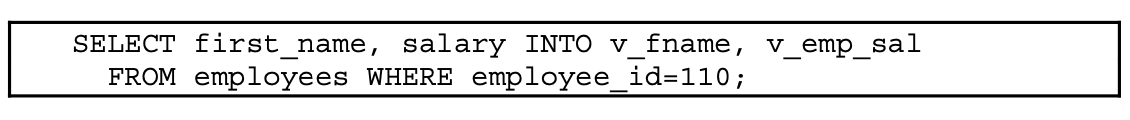
**v\_pf\_percent NUMBER := 12;**

**v\_fname VARCHAR2(15);**

**v\_emp\_sal NUMBER(10);**

Изображение выглядит как текст

Автоматически созданное описание

d) Include the following SQL statement in the executable section: Изображение выглядит как текст

Автоматически созданное описание

e) Change the line that prints "Hello World" to print "Hello" and the first name. Then, comment the lines that display the dates and print the bind variables.

**BEGIN**

**v\_tomorrow := v\_today + 1;**

**/\***

**:b\_basic\_percent := 45;**

**:b\_pf\_percent := 12;**

**\*/**

**SELECT first\_name, salary INTO v\_fname, v\_emp\_sal**

**FROM employees WHERE employee\_id = 110;**

**DBMS\_OUTPUT.PUT\_LINE('Hello' || v\_fname);**

**--DBMS\_OUTPUT.PUT\_LINE('Today''s date: ' || v\_today);**

**--DBMS\_OUTPUT.PUT\_LINE('Tomorrow''s date: ' || v\_tomorrow);**

**END;**

**--/**

**--PRINT b\_basic\_percent;**

**--PRINT b\_pf\_percent;**

**Изображение выглядит как текст

Автоматически созданное описание**

f) Calculate the contribution of the employee towards provident fund (PF). PF is 12% of the basic salary, and the basic salary is 45% of the salary. Use local variables for the calculation. Try to use only one expression to calculate the PF. Print the employee’s salary and his or her contribution toward PF.

**SET SERVEROUTPUT ON**

**--VARIABLE b\_basic\_percent NUMBER;**

**--VARIABLE b\_pf\_percent NUMBER;**

**DECLARE**

**v\_today DATE := SYSDATE;**

**v\_tomorrow v\_today%TYPE;**

**v\_basic\_percent NUMBER := 45;**

**v\_pf\_percent NUMBER := 12;**

**v\_fname VARCHAR2(15);**

**v\_emp\_sal NUMBER(10);**

**BEGIN**

**v\_tomorrow := v\_today + 1;**

**/\***

**:b\_basic\_percent := 45;**

**:b\_pf\_percent := 12;**

**\*/**

**SELECT first\_name, salary INTO v\_fname, v\_emp\_sal**

**FROM employees WHERE employee\_id = 110;**

**DBMS\_OUTPUT.PUT\_LINE('Hello ' || v\_fname);**

**DBMS\_OUTPUT.PUT\_LINE('Your salary: ' || v\_emp\_sal);**

**DBMS\_OUTPUT.PUT\_LINE('Your contribution toward PF: '**

**|| v\_emp\_sal\*v\_basic\_percent/100\*v\_pf\_percent/100);**

**--DBMS\_OUTPUT.PUT\_LINE('Today''s date: ' || v\_today);**

**--DBMS\_OUTPUT.PUT\_LINE('Tomorrow''s date: ' || v\_tomorrow);**

**END;**

**--/**

**--PRINT b\_basic\_percent;**

**--PRINT b\_pf\_percent;**

g) Execute and save your script as lab\_03\_03\_soln.sql. The sample output is as follows:

Изображение выглядит как стол

Автоматически созданное описание