 



Database Programming with PL/SQL 2-6: Nested Blocks and Variable Scope Practice Activities

# Vocabulary

Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| Qualifier | A name given to a block of code which allows access to the variables that have scope, but are not visible. |
| Blocuri imbricate | Consists of all the blocks in which the variable is either local (the declaring block) or global (nested blocks within the declaring block) . |
| Visibility | The portion of the program where the variable can be accessed without using a qualifier. |

# Try It / Solve It

1. Evaluate the PL/SQL block below and determine the value of each of the following variables according to the rules of scoping.

DECLARE

weight NUMBER(3) := 600;

message VARCHAR2(255) := 'Product 10012'; BEGIN

DECLARE

weight NUMBER(3) := 1;

message VARCHAR2(255) := 'Product 11001'; new\_locn VARCHAR2(50) := 'Europe';

BEGIN

weight := weight + 1; 🡪 1 + 1 = 2

new\_locn := 'Western ' || new\_locn; 🡪Western Europe

-- Position 1 -- END;

weight := weight + 1; 🡪 600 + 1 = 601

message := message || ' is in stock'; 🡪Product 10012 is in stock

-- Position 2 -- END;

* 1. The value of weight at position 1 is: 2
  2. The value of new\_locn at position 1 is: Western Europe
  3. The value of weight at position 2 is: 601
  4. The value of message at position 2 is: Product 10012 is in stock
  5. The value of new\_locn at position 2 is: NULL

1. Enter and run the following PL/SQL block, which contains a nested block. Look at the output and answer the questions.

DECLARE

v\_employee\_id employees.employee\_id%TYPE; v\_job employees.job\_id%TYPE;

BEGIN

SELECT employee\_id, job\_id INTO v\_employee\_id, v\_job FROM employees

WHERE employee\_id = 100;

DECLARE

v\_employee\_id employees.employee\_id%TYPE; v\_job employees.job\_id%TYPE;

BEGIN

SELECT employee\_id, job\_id INTO v\_employee\_id, v\_job FROM employees

WHERE employee\_id = 103; DBMS\_OUTPUT.PUT\_LINE(v\_employee\_id || ' is a(n) ' || v\_job);

END;

DBMS\_OUTPUT.PUT\_LINE(v\_employee\_id || ' is a(n) ' || v\_job); END;

* 1. Why does the inner block display the job\_id of employee 103, not employee 100?

Pentru ca in clauza WHERE din blocul interior am impus ca employee\_id sa fie 103 .

* 1. Why does the outer block display the job\_id of employee 100, not employee 103?

Pentru ca deja sunt in blocul exterior in care clauza WHERE imi impune ca employee\_id sa fie 100 .

* 1. Modify the code to display the details of employee 100 in the inner block. Use block labels.

<<outer>>

DECLARE

v\_employee\_id employees.employee\_id%TYPE; v\_job employees.job\_id%TYPE;

BEGIN

SELECT employee\_id, job\_id INTO v\_employee\_id, v\_job FROM employees

WHERE employee\_id = 100;

<<inner>>

DECLARE

v\_employee\_id employees.employee\_id%TYPE; v\_job employees.job\_id%TYPE;

BEGIN

SELECT employee\_id, job\_id INTO v\_employee\_id, v\_job FROM employees

WHERE employee\_id = 100; DBMS\_OUTPUT.PUT\_LINE(inner.v\_employee\_id || ' is a(n) ' || v\_job);

END;

DBMS\_OUTPUT.PUT\_LINE(outer.v\_employee\_id || ' is a(n) ' || v\_job); END;

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.