www.oracle.com/academy





**Escola del Clot**

CFGS Desenvolupament d'Aplicacions Web

CFGS Desenvolupament d'Aplicacions Multiplataforma

jaume.reixach@fje.edu

**M02. Bases de dades UF4 BBDD objecte-relacional**

**Activitat 2.6**

Database Programming with PL/SQL

2-6: Nested Blocks and Variable Scope

Practice Activities

**Vocabulary**

Identify the vocabulary word for each definition below.

A name given to a block of code which allows access to the variables that have scope, but are not visible.

Calificador

Consists of all the blocks in which the variable is either local (the declaring block) or global (nested blocks within the declaring block) .

Variables scope

The portion of the program where the variable can be accessed without using a qualifier.

Variables visibles

**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;

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

-- Position 1 --

END;

weight := weight + 1;

message := message || ' is in stock';

-- Position 2 --

END;

1. The value of weight at position 1 is: 2
   1. The value of new\_locn at position 1 is: Western europe
   2. The value of weight at position 2 is: 601
   3. The value of message at position 2 is: Product 10012 is in stock
   4. The value of new\_locn at position 2 is: Out of range
2. 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?

Porque las dos declaraciones v\_job estan en el scope y el inner block

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

Porque la declaracions del inner block esta fuera del scope del block exterior

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

<<outer\_block>>

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\_block>>

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(outer\_block.v\_employee\_id||

‘ is a '||outer\_block.v\_job);

END;

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

END;

Copyright © 2016, 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.