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

JESUÏTES El Clot
Escola del Clot

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

Vocabulary

Identify the vocabulary word for each definition below.

Calificador	A name given to a block of code which allows access to the variables that have scope, but are not visible.
Variables scope	Consists of all the blocks in which the variable is either local (the declaring block) or global (nested blocks within the declaring block) .
Variables visibles	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;
    new_locn := 'Western ' || new_locn;
    -- Position 1 --
  END;

  weight := weight + 1;
  message := message || ' is in stock';
  -- Position 2 --
END;
```

A. The value of weight at position 1 is: 2
C. The value of new_locn at position 1 is: Western europe
D. The value of weight at position 2 is: 601
E. The value of message at position 2 is: Product 10012 is in stock
F. 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;
```

A. 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

B. 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

C. 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;
```