Database Programming with PL/SQL

8-3: Passing Parameters

Practice Activities

Vocabulary

Identify the vocabulary word for each definition below:

OUT PARAMETER	Returns a value to the caller
IN PARAMETER	Provides values for a subprogram to process
NAMED NOTATION	Lists the actual parameters in arbitrary order and uses the association operator ('=>' which is an equal and an arrow together) to associate a named formal parameter with its actual parameter
COMBINATION NOTATION	Lists some of the actual parameters as positional (no special operator) and some as named (with the => operator)
POSITIONAL NOTATION	Lists the actual parameters in the same order as the formal parameters
IN OUT PARAMETER	Supplies an input value, which may be returned as a modified value

Try It / Solve It

1. Name the three modes for parameters and indicate which mode is the default mode. Which mode cannot be modified inside the procedure?

Posición, nombre y combinado, el que es por posición es por defecto

2. Procedures:

A. Create a procedure that receives a country_id as an IN parameter and returns the name and population of that country as OUT parameters. Include an exception handler to trap the NO_DATA_FOUND exception if the country does not exist. The procedure should not display the returned values; this will be done in the next step. Name your procedure find_area_pop. Save your code.

```
CREATE OR REPLACE PROCEDURE find_area_pop(
p_id IN countries.country_id%type,
v_name OUT countries.country_name%type,
v_population OUT countries.population%type)
AS
BEGIN
SELECT country_name, population INTO v_name, v_population FROM COUNTRIES WHERE country_id=p_id;
END;
```

B. Test your procedure by creating and executing an anonymous block which invokes the procedure and displays the returned OUT values. Save your code. Run the block twice, with country_ids 2 (Canada) and 10 (does not exist).

```
 DECLARE
v_name countries.country_name%type;
v_population countries.population%type;
v_id countries.country_id%type:=2;
BEGIN
find_area_pop(v_id, v_name, v_population);
dbms_output.put_line(v_name || ' '|| v_population);
END;
```

C. Retrieve your procedure code and modify it to add a third OUT parameter which is the population density of the country, using the formula: density = (population / area). You will need to modify your SELECT statement to fetch the area column value into a local variable. Save your modified code.

```
CREATE OR REPLACE PROCEDURE find_area_pop(
p_id IN countries.country_id%type,
v_name OUT countries.country_name%type,
v_population OUT countries.population%type,
V_densidad OUT NUMBER)
AS
```

```
v_area countries.area%type;

BEGIN

SELECT country_name, population,area INTO v_name, v_population, v_area FROM COUNTRIES WHERE country_id=p_id;

v_densidad:= (v_population/v_area);

END;
```

D. Test your modifed procedure using country_id 2. You will need to modify your calling anonymous block to declare and pass a fourth actual parameter to receive the population density from the procedure. Save your code.

```
DECLARE

v_name countries.country_name%type;

v_population countries.population%type;

v_id countries.country_id%type:=2;

v_densidad NUMBER(5,2);

BEGIN

find_area_pop(v_id, v_name, v_population,v_densidad);

dbms_output.put_line(v_name || ' '|| v_population || ' '||v_densidad );

END;
```

3. Create a procedure which accepts an integer as an IN OUT parameter and returns the square of that integer, for example the square of 4 is 16. Save your code. Test your procedure from an anonymous block three times, using integer values 4, 7, and –20 (negative 20).

```
CREATE OR REPLACE PROCEDURE square(
p_var IN OUT NUMBER
) AS

begin
p_var:= p_var*p_var;
end;

DECLARE
var NUMBER:=6;
begin
square(var);
DBMS_OUTPUT.PUT_LINE(var);
END;
```

4. List the three methods of passing parameters to a procedure.

IN, OUT, IN OUT.

A. Retrieve your anonymous block from question 2D and modify its call to find_area_pop to pass the four parameters using named notation. Test your block, again using country_id 2 (Canada).

If you have forgotten the p_ names of the procedure's formal parameters, how can you refresh your memory?

DESC find_area_pop;

B. Modify the anonymous block from the previous step to pass the FIRST two parameters using named notation and the LAST two using positional notation. Test the block again. What happens?

No funciona, porque al usar los parámetros por nombre se pierde la posicion

C. Correct the problem in the previous step by modifying the anonymous block again to pass the first two parameters using positional notation and the last two using named notation. Test the block again.

```
DECLARE

v_name countries.country_name%type;

v_population countries.population%type;

v_id countries.country_id%type:=2;

v_densidad NUMBER(5,2);

BEGIN

find_area_pop(v_id, v_name,v_population=>v_population, v_densidad=>v_densidad);

dbms_output.put_line(v_name || ' '|| v_population || ' '||v_densidad );

END;
```

5. In your own words, describe the purpose of the DEFAULT option for parameters and state the two syntax options for providing the default value in the procedure header.

    Los parametros son variables que recibe un procedimiento, son útiles porque se evita tener repetir código para adaptarlo a cada caso.

    Existen dos formas de asignar un valor por defecto a un parámetro

    P_var NUMBER := 10;

    P_var NUMBER :=DEFAULT 1400;

6. Find the country_id of your own country by executing a suitable SELECT…FROM countries…. Then retrieve your find_area_pop procedure from question 2C. Modify the code to use your country_id as a default value for the country_id IN parameter. Save your code. Then retrieve your anonymous block from question 2D and modify it so that it does NOT pass the country_id to the procedure. Test the block and check that your country's details are returned and displayed. If your modified anonymous block does not work, correct it so it will.

```
CREATE OR REPLACE PROCEDURE find_area_pop(

p_id IN out countries.country_id%type DEFAULT 52,

v_name OUT countries.country_name%type,

v_population OUT countries.population%type,

V_densidad OUT NUMBER)

AS

v_area countries.area%type;

BEGIN

SELECT country_name, population,area INTO v_name, v_population, v_area FROM
COUNTRIES WHERE country_id=p_id;

v_densidad:= (v_population/v_area);

END;
```

```
DECLARE

v_name countries.country_name%type;

v_population countries.population%type;

v_id countries.country_id%type:=2;

v_densidad NUMBER(5,2);

BEGIN

find_area_pop(v_name=>v_name,
v_population=>v_population,v_densidad=>v_densidad);

dbms_output.put_line(v_name || ' '|| v_population || ' '||v_densidad );

END;
```