

Database Programming with PL/SQL

8-2: Using Parameters in Procedures

Practice Activities

# Vocabulary

Identify the vocabulary word for each definition below:

|  |  |
| --- | --- |
| **Parametru** | Pass or communicate data between the caller and subprogram. |
| **Argument** | The actual value assigned to a parameter. |
| **Parametru actual** | Can be literal values, variables, or expressions that are provided in the parameter list of a called subprogram. |
| **Parametru formal** | A parameter name declared in the procedure heading. |

# Try It / Solve It

1. In your own words, describe parameters and the purpose they serve in PL/SQL subprograms.

Parametrul este o formă de variabilă specială utilizată în antetul subprogramelor.

1. Using the COUNTRIES table:

* 1. Create a procedure that accepts a country\_id as a parameter and displays the name of the country and its capitol city. Name your procedure get\_country\_info. Save your procedure definition for later use.

CREATE OR REPLACE procedure get\_country\_info

(p\_country\_id IN countries.country\_id%TYPE,

p\_capitol OUT countries.capitol%TYPE,

p\_country\_name OUT countries.country\_name%TYPE)

IS

BEGIN

SELECT country\_name, capitol INTO p\_country\_name, p\_capitol

FROM countries

WHERE country\_id = p\_country\_id;

DBMS\_OUTPUT.PUT\_LINE('Country\_name:' ||p\_country\_name ||' '||p\_capitol);

END get\_country\_info;

* 1. Execute your procedure from an anonymous block, using country\_id 90.

DECLARE

p\_country\_name countries.country\_name%TYPE;

p\_capitol countries.capitol%TYPE;

BEGIN

get\_country\_info(90, p\_country\_name, p\_capitol);

END;

* 1. Re-execute the procedure from the anonymous block, this time using country\_id 95. What happens?

**ORA-01403: no data found**

* 1. Retrieve your procedure code from Saved SQL and modify it to trap the NO\_DATA\_FOUND exception in an exception handler. Execute the modified procedure using country\_id 95 again. Now what happens?

CREATE OR REPLACE procedure get\_country\_info

(p\_country\_id IN countries.country\_id%TYPE,

p\_capitol OUT countries.capitol%TYPE,

p\_country\_name OUT countries.country\_name%TYPE)

IS

BEGIN

SELECT country\_name, capitol INTO p\_country\_name, p\_capitol

FROM countries

WHERE country\_id = p\_country\_id;

DBMS\_OUTPUT.PUT\_LINE('Country\_name:' ||p\_country\_name ||' '||p\_capitol);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Exception NO\_DATA\_FOUND');

END get\_country\_info;

DECLARE

p\_country\_name countries.country\_name%TYPE;

p\_capitol countries.capitol%TYPE;

BEGIN

get\_country\_info(95, p\_country\_name, p\_capitol);

END;

In acest caz apare mesajul pe care l-am pus sa se afiseza atunci cand imi apare exceptia, adica ‚Exception NO\_DATA\_FOUND’ .

1. In your own words, describe what a formal parameter is and what an actual parameter is. Also, name three variations for an actual parameter.

Un parametru formal este cel declarat in heading-ul procedurii. Corespondenta nume-valoare in calling enviroment se numeste parametru actual .

Parametrii actuali pot fi litearal values, variables or expressions.

.

1. Procedure Exercise:

* 1. Write a procedure that displays the number of countries in a given region whose highest elevations exceed a given value. The procedure should accept two formal parameters, one for a region\_id and the other for an elevation value for comparison. Use DBMS\_OUTPUT.PUT\_LINE to display the results in a message. Save your procedure code.

CREATE OR REPLACE PROCEDURE ncountries

(p\_region\_id countries.region\_id%TYPE,

p\_elevation countries.highest\_elevation%TYPE)

IS

v\_count NUMBER(6);

BEGIN

SELECT COUNT(\*) INTO v\_count

FROM countries

WHERE region\_id = p\_region\_id AND highest\_elevation > p\_elevation;

DBMS\_OUTPUT.PUT\_LINE ('Region id:' || p\_region\_id ||' '||'elevation' || p\_elevation ||' numbers:' ||v\_count); END;

* 1. Execute your procedure using the value 5 for the region\_id and 2000 for the highest elevation.

BEGIN

ncountries(5, 2000);

END;

* 1. DESCRIBE your procedure to check the names and datatypes of its formal parameters.

DESCRIBE ncountries;

P\_region\_id, p\_number sunt de tip IN, NUMBER .

* 1. Retrieve your procedure code from Saved SQL and modify it to accept a third formal parameter of datatype CHAR. The procedure should display a count of the number of countries in a given region whose highest elevations exceed a given value and whose country name starts with a given alphabetic character. Your SELECT statement should include a WHERE condition to compare the first character of each country’s name with the third parameter value (Hint: use SUBSTR). Save your work again and DESCRIBE the modified procedure.

CREATE OR REPLACE PROCEDURE ncountries

(p\_region\_id countries.region\_id%TYPE,

p\_elevation countries.highest\_elevation%TYPE,

p\_char CHAR)

IS

v\_count NUMBER(6);

BEGIN

SELECT COUNT(\*) INTO v\_count

FROM countries

WHERE region\_id = p\_region\_id AND highest\_elevation > p\_elevation AND SUBSTR(country\_name, 1, 1) = p\_char;

DBMS\_OUTPUT.PUT\_LINE ('Region id:' || p\_region\_id ||' '||'elevation' || p\_elevation ||' numbers:' ||v\_count ||' first letter ' || p\_char);

END;

DESCRIBE ncountries;

* 1. Write an anonymous block which declares three variables to store actual parameter values for the region\_id, elevation, and area, and then executes the procedure passing these values. Execute the block using values 5, 2000, and ‘B’.

DECLARE

v\_region\_id countries.region\_id%TYPE := 5;

v\_elevation countries.highest\_elevation%TYPE := 2000;

v\_area CHAR(1) := 'B';

BEGIN

ncountries(v\_region\_id, v\_elevation, v\_area);

END;

* 1. Modify your anonymous block to use the same actual parameter values but pass them to the procedure in a different order: (5, ‘B’, 2000). Execute the block. What happens and why?

DECLARE

v\_region\_id countries.region\_id%TYPE := 5;

v\_elevation countries.highest\_elevation%TYPE := 2000;

v\_area CHAR(1) := 'B';

BEGIN

ncountries(v\_region\_id, v\_area, v\_elevation);

END;

**ORA-06502: PL/SQL: numeric or value error: character to number conversion error**

**Apare o eroare din cauza faptului ca a primit un parametru de alt tip decat cel asteptat.**