8 3

Agustín Alejandro Mota Hinojosa

December 3, 2023

Contents

1 Vocabulary

1. **Function:** Returns a value to the caller.
2. **Input Parameter:** Provides values for a subprogram to process.
3. **Named Parameter Association:** 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.
4. **Mixed Parameter Association:** Lists some of the actual parameters as positional (no special operator) and some as named (with the ‘=>‘ operator).
5. **Positional Parameter Association:** Lists the actual parameters in the same order as the formal parameters.
6. **Input/Output Parameter:** Supplies an input value, which may be returned as a modified value.

2 Try

1. Modes for Parameters:
   - IN Mode: Default mode. It passes values into the subprogram.
   - OUT Mode: It returns values from the subprogram.
   - IN OUT Mode: It both passes values into and returns values from the subprogram.
2. Procedures
   (a) code:

```
CREATE OR REPLACE PROCEDURE find_area_pop(
   p_country_id IN COUNTRIES.COUNTRY_ID%TYPE,
   p_country_name OUT COUNTRIES.COUNTRY_NAME%TYPE,
   p_population OUT COUNTRIES.POPULATION%TYPE,
   p_density OUT NUMBER
) AS
   v_area COUNTRIES.AREA%TYPE;
BEGIN
   SELECT COUNTRY_NAME, POPULATION, AREA
```

```
   INTO p_country_name, p_population, v_area
   FROM COUNTRIES
   WHERE COUNTRY_ID = p_country_id;

   p_density := p_population / v_area;
EXCEPTION
   WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('Country ID ' || p_country_id || ' not found.');
END find_area_pop;
/
```

(b) Test:

```
DECLARE
   v_country_id NUMBER;
   v_country_name COUNTRIES.COUNTRY_NAME%TYPE;
   v_population COUNTRIES.POPULATION%TYPE;
   v_density NUMBER;
BEGIN
   v_country_id := 2; -- Canada
   find_area_pop(v_country_id, v_country_name, v_population, v_density);
   DBMS_OUTPUT.PUT_LINE('Country Name: ' || v_country_name);
   DBMS_OUTPUT.PUT_LINE('Population: ' || v_population);
   DBMS_OUTPUT.PUT_LINE('Density: ' || v_density);

   v_country_id := 10; -- Non-existent country
   find_area_pop(v_country_id, v_country_name, v_population, v_density);
END;
```

(c) modify:

```
CREATE OR REPLACE PROCEDURE find_area_pop(
   p_country_id IN COUNTRIES.COUNTRY_ID%TYPE,
   p_country_name OUT COUNTRIES.COUNTRY_NAME%TYPE,
   p_population OUT COUNTRIES.POPULATION%TYPE,
   p_density OUT NUMBER
) AS
   v_area COUNTRIES.AREA%TYPE;
BEGIN
   SELECT COUNTRY_NAME, POPULATION, AREA
   INTO p_country_name, p_population, v_area
   FROM COUNTRIES
   WHERE COUNTRY_ID = p_country_id;

   p_density := p_population / v_area;
EXCEPTION
   WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('Country ID ' || p_country_id || ' not found.');
END find_area_pop;
/
```

(d) Test modified:

```
DECLARE
   v_country_id NUMBER;
   v_country_name COUNTRIES.COUNTRY_NAME%TYPE;
```

```
   v_population COUNTRIES.POPULATION%TYPE;
   v_density NUMBER;
BEGIN
   v_country_id := 2;
   find_area_pop(p_country_id => v_country_id,
                 p_country_name => v_country_name,
                 p_population => v_population,
                 p_density => v_density);
   DBMS_OUTPUT.PUT_LINE('Country Name: ' || v_country_name);
   DBMS_OUTPUT.PUT_LINE('Population: ' || v_population);
   DBMS_OUTPUT.PUT_LINE('Density: ' || v_density);
END;
```

3. Create Procedure for Squaring an Integer:

```
CREATE OR REPLACE PROCEDURE square_of_integer(
   p_number IN OUT NUMBER
) AS
BEGIN
   p_number := p_number * p_number;
END square_of_integer;
/
```

   (a) Test

```
DECLARE
   v_number NUMBER := 4;
BEGIN
   square_of_integer(v_number);
   DBMS_OUTPUT.PUT_LINE('Square of 4: ' || v_number);

   v_number := 7;
   square_of_integer(v_number);
   DBMS_OUTPUT.PUT_LINE('Square of 7: ' || v_number);

   v_number := -20;
   square_of_integer(v_number);
   DBMS_OUTPUT.PUT_LINE('Square of -20: ' || v_number);
END;
```

4. Methods of Passing Parameters:

   (a) Retrieve Anonymous Block:

```
DECLARE
   v_country_id NUMBER := 2;
   v_country_name COUNTRIES.COUNTRY_NAME%TYPE;
   v_population COUNTRIES.POPULATION%TYPE;
   v_density NUMBER;
BEGIN
   find_area_pop(p_country_id, v_country_name, v_population, v_density);
   DBMS_OUTPUT.PUT_LINE('Country Name: ' || v_country_name);
   DBMS_OUTPUT.PUT_LINE('Population: ' || v_population);
   DBMS_OUTPUT.PUT_LINE('Density: ' || v_density);
END;
```

(b) Modify Anonymous Block:

```
plsql

DECLARE
   v_country_id NUMBER := 2;
   v_country_name COUNTRIES.COUNTRY_NAME%TYPE;
   v_population COUNTRIES.POPULATION%TYPE;
   v_density NUMBER;
BEGIN
   find_area_pop(v_country_name => v_country_name,
                 v_population => v_population,
                 v_density => v_density,
                 p_country_id => v_country_id);
   DBMS_OUTPUT.PUT_LINE('Country Name: ' || v_country_name);
   DBMS_OUTPUT.PUT_LINE('Population: ' || v_population);
   DBMS_OUTPUT.PUT_LINE('Density: ' || v_density);
END;
```