

Database Programming with PL/SQL

4-4: Iterative Control: WHILE and FOR Loops

Practice Activities

Vocabulary

Identify the vocabulary word for each definition below:

WHILE Loop	Repeats a sequence of statements until the controlling condition is no longer TRUE.
FOR Loop	Repeats a sequence of statements until a set number of iterations have been completed.

Try It / Solve It

1. Write a PL/SQL block to display the country_id and country_name values from the COUNTRIES table for country_id whose values range from 51 through 55. Use a WHILE loop. Increment a variable from 51 through 55. Test your variable to see when it reaches 55. EXIT the loop after you have displayed the 5 countries.

Folosete o variabil v_country_id care este incrementat manual de la 51 la 55. Codul afieaz numele i ID-ul rii folosind DBMS_OUTPUT.PUT_LINE.

2. Write a PL/SQL block to display the country_id and country_name values from the COUNTRIES table for country_id whose values range from 51 through 55 *in the reverse order*. Use a FOR loop.

Itereaz automat în ordine invers între valorile 55 i 51. Nu necesit incrementare manual, deoarece ciclul FOR se ocup de aceasta.

3. Execute the following statements to build a new_emps table.

```
DROP TABLE new_emps;
```

```
CREATE TABLE new_emps AS SELECT * FROM employees;
```

```
ALTER TABLE new_emps ADD stars VARCHAR2(50);
```

1. DECLARE

```
v_country_id NUMBER := 51; -- Variabil pentru incrementare
v_country_name VARCHAR2(100); -- Variabil pentru numele rii
```

BEGIN

```
WHILE v_country_id <= 55 LOOP
    -- Obine numele rii pentru fiecare ID
    SELECT country_name INTO v_country_name
    FROM countries
    WHERE country_id = v_country_id;
```

-- Afieaz valorile

```
DBMS_OUTPUT.PUT_LINE('Country ID: ' || v_country_id || ', Country Name: ' || v_country_name);
```

```
v_country_id := v_country_id + 1;
END LOOP;
```

2. DECLARE

```
v_country_name VARCHAR2(100); -- Variabil pentru numele rii
BEGIN
```

```
FOR v_country_id IN REVERSE 51..55 LOOP
```

-- Obine numele rii pentru fiecare ID

```
SELECT country_name INTO v_country_name
FROM countries
```

```
WHERE country_id = v_country_id;
```

-- Afieaz valorile

```
DBMS_OUTPUT.PUT_LINE('Country ID: ' ||
v_country_id || ', Country Name: ' || v_country_name);
```

```
END LOOP;
END;
```

- A. Create a PL/SQL block that inserts an asterisk in the stars column for every whole \$1,000 of an employee's salary. For example, if an employee has salary of \$7,800, the string "*****" would be inserted, and, if an employee has salary of \$3,100, the string "***" would be inserted. Use the following code as a starting point.

```

DECLARE
  v_empno          new_emps.employee_id%TYPE := <employee_id>;
  v_asterisk       new_emps.stars%TYPE := NULL;
  v_sal_in_thousands new_emps.salary%TYPE;
BEGIN
  SELECT NVL(TRUNC(salary/1000), 0) INTO v_sal_in_thousands
    FROM new_emps WHERE employee_id = v_empno;
  FOR ...
  ...
  UPDATE new_emps
    SET stars = v_asterisk
    WHERE employee_id = v_empno;
END;

```

- B. Test your code using employee_ids 124 and 142, then confirm the results.

```
SELECT employee_id, stars FROM new_emps WHERE employee_id IN (124, 142);
```

Pentru angajaii cu ID-urile:

Employee 124:

Salariu: \$7,800 ir de asteriscuri: *****.

Employee 142:

Salariu: \$3,100 ir de asteriscuri: ***.

3.A Explicaii:

Iniializare variabile:

v_empno: Stocheaz ID-ul angajatului curent în bucl.

v_asterisk: Va stoca irul de asteriscuri calculat.

v_sal_in_thousands: Este determinat prin împrirea salariului la 1000 i trunchierea rezultatului la întreaga parte.

Bucula FOR: Itereaz peste angajaii cu ID-urile specificate (124 i 142).

Calculul irului de asteriscuri:

RPAD('*', v_sal_in_thousands, '*') construiește un ir de asteriscuri cu lungimea egal cu valoarea salariului în mii.

Instruciunea UPDATE: Actualizeaz coloana stars cu irul generat.

Afiarea rezultatelor: DBMS_OUTPUT.PUT_LINE este utilizat pentru verificarea ieirii.