 



# Database Programming with PL/SQL

4-4: Iterative Control: WHILE and FOR Loops

# Practice Activities

## Vocabulary

Identify the vocabulary word for each definition below:

|  |  |
| --- | --- |
| Bucla WHILE | Repeats a sequence of statements until the controlling condition is no longer TRUE. |
| Bucla FOR | 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.

DECLARE

v\_country\_id countries.country\_id%TYPE := 51;

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

v\_counter NUMBER(2) := 51;

BEGIN

WHILE v\_counter <= 55 LOOP

SELECT country\_id, country\_name INTO v\_country\_id, v\_country\_name

FROM countries

WHERE country\_id = v\_counter;

v\_counter := v\_counter + 1;

DBMS\_OUTPUT.PUT\_LINE('Country\_id: '||v\_country\_id ||' country\_name: '|| v\_country\_name);

END LOOP;

END;

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 *in the reverse order*. Use a FOR loop.

DECLARE

v\_country\_id countries.country\_id%TYPE := 51;

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

v\_counter NUMBER(2) := 51;

BEGIN

FOR v\_counter IN REVERSE 51..55 LOOP

SELECT country\_id, country\_name INTO v\_country\_id, v\_country\_name

FROM countries

WHERE country\_id = v\_counter;

DBMS\_OUTPUT.PUT\_LINE('Country\_id: '||v\_country\_id ||' country\_name: '|| v\_country\_name);

END LOOP;

END;

1. 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. 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;

* 1. Test your code using employee\_ids 124 and 142, then confirm the results.

DECLARE

v\_empno new\_emps.employee\_id%TYPE := 124;

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 i IN 1..v\_sal\_in\_thousands LOOP

v\_asterisk := v\_asterisk || '\*';

END LOOP;

UPDATE new\_emps SET stars = v\_asterisk

WHERE employee\_id = v\_empno;

END;

DECLARE

v\_empno new\_emps.employee\_id%TYPE := 142;

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 i IN 1..v\_sal\_in\_thousands LOOP

v\_asterisk := v\_asterisk || '\*';

END LOOP;

UPDATE new\_emps SET stars = v\_asterisk

WHERE employee\_id = v\_empno;

END;

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.