Database Programming with PL/SQL

4-4: Iterative Control: WHILE and FOR Loops

Practice Activities

For *counter* in [Reverse] lower bound..higher bound LOOP

statement1

statement…

Nu declarati counter. Se declara implicit!

Chiar daca zici reverse, lower bound ramane tot (sa zic 1), si higher bound ramane tot (sa zic n)

Vocabulary

|  |  |
| --- | --- |
| WHILE | Repeats a sequence of statements until the controlling condition  is no longer TRUE. |
| 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\_id countries.country\_id%TYPE;

v\_name countries.country\_name%TYPE;

BEGIN

v\_id:=51;

while v\_id < 56 LOOP

select country\_name into v\_name from countries where country\_id=v\_id;

DBMS\_OUTPUT.PUT\_LINE(v\_id||’ ‘||v\_country);

end loop;

END;

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.

Declare

v\_id countries.country\_id%TYPE;

v\_name countries.country\_name%TYPE;

BEGIN

for counter in reverse range 51..55 LOOP

v\_id:=counter;

select country\_name into v\_name from countries where country\_id=v\_id;

DBMS\_OUTPUT.PUT\_LINE(v\_id||’ ‘||v\_country);

end loop;

END;

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);

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 I in 1..v\_sal\_in\_thousands LOOP

v\_asterisk:=concat(v\_asterisk, ‘\*’);

end loop;

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.