Database Programming with PL/SQL

5-3: Cursor FOR Loops

Practice Activities

Vocabulary

Automates standard cursor-handling operations such as

OPEN, FETCH, %NOTFOUND, and CLOSE so that they

do not need to be coded explicitly – Cursor FOR loop

Try It / Solve It

1. Describe two benefits of using a cursor FOR loop.

OPEN FETCH si CLOSE au loc automat

variabila record se declara implicit in FOR

mai scurt si mai usor de citit

2. Modify the following PL/SQL block so that it uses a cursor FOR loop. Keep the explicit cursor

declaration in the DECLARE section. Test your changes.

DECLARE

CURSOR countries\_cur IS

SELECT country\_name, national\_holiday\_name, national\_holiday\_date

FROM countries

WHERE region\_id = 5;

countries\_rec countries\_cur%ROWTYPE;

BEGIN

OPEN countries\_cur;

LOOP

FETCH countries\_cur INTO countries\_rec;

EXIT WHEN countries\_cur%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE ('Country: ' || countries\_rec.country\_name

|| ' National holiday: '|| countries\_rec.national\_holiday\_name

|| ', held on: '|| countries\_rec.national\_holiday\_date);

END LOOP;

CLOSE countries\_cur;

END;

DECLARE

 CURSOR countries\_cur IS

 SELECT country\_name, national\_holiday\_name, national\_holiday\_date

 FROM wf\_countries

 WHERE region\_id = 5;

 countries\_rec countries\_cur%ROWTYPE;

BEGIN

 FOR countries\_rec IN countries\_cur LOOP

 DBMS\_OUTPUT.PUT\_LINE ('Country: ' || countries\_rec.country\_name || ' National holiday: '|| countries\_rec.national\_holiday\_name || ', held on: '|| countries\_rec.national\_holiday\_date);

 END LOOP;

END;

3. Modify your answer to question 2 to declare the cursor using a subquery in the FOR…LOOP

statement, rather than in the declaration section. Test your changes again.

BEGIN

 FOR countries\_rec IN (SELECT country\_name, national\_holiday\_name, national\_holiday\_date FROM wf\_countries WHERE region\_id = 5) LOOP

 DBMS\_OUTPUT.PUT\_LINE ('Country: ' || countries\_rec.country\_name || ' National holiday: '|| countries\_rec.national\_holiday\_name || ', held on: '|| countries\_rec.national\_holiday\_date);

 END LOOP;

END;

4. Using the COUNTRIES table, write a cursor that returns countries with a highest\_elevation greater than 8,000 m. For each country, display the country\_name, highest\_elevation, and climate. Use a cursor FOR loop, declaring the cursor using a subquery in the FOR…LOOP statement.

BEGIN

 FOR countries\_rec IN (SELECT country\_name, highest\_elevation, climate FROM wf\_countries WHERE highest\_elevation > 8000) LOOP

 DBMS\_OUTPUT.PUT\_LINE (countries\_rec.country\_name || ' '|| countries\_rec.highest\_elevation || ' '|| countries\_rec.climate);

 END LOOP;

END;

5. This question uses a join of the SPOKEN\_LANGUAGES and COUNTRIES tables with a GROUP

BY and HAVING clause.

Write a PL/SQL block to fetch and display all the countries that have more than six spoken

languages. For each such country, display country\_name and the number of spoken languages.

Use a cursor FOR loop, but declare the cursor explicitly in the DECLARE section. After all the

rows have been fetched and displayed, display an extra row showing the total number of countries

having more than six languages. (Hint: Declare a variable to hold the value of %ROWCOUNT.)

DECLARE

 v\_count NUMBER(4);

 CURSOR languages\_cursor IS SELECT country\_name, COUNT(\*) AS number\_countries

 FROM wf\_countries c, wf\_spoken\_languages sl

 WHERE c.country\_id = sl.country\_id

 GROUP BY country\_name HAVING COUNT(\*) >6;

BEGIN

 FOR languages\_rec IN languages\_cursor

 LOOP

 DBMS\_OUTPUT.PUT\_LINE(languages\_rec.country\_name || ' ' || languages\_rec.number\_countries);

 v\_count := languages\_cursor%ROWCOUNT;

 END LOOP;

 DBMS\_OUTPUT.PUT\_LINE(v\_count);

END;

6. Why did your block in question 4 need to declare the cursor explicitly, instead of declaring it as a

subquery in the FOR…LOOP statement?

Because we need the atribute %ROWCOUNT that can't be used with a subquery, but can be used on a cursor.