Database Programming with PL/SQL

5-4: Cursors with Parameters

Practice Activities

Vocabulary

No new vocabulary for this lesson

Try It / Solve It

1. Describe the benefit of using one or more parameters with a cursor.

Instead of using more cursors with different values, we can use the same cursor by creating a parameter that we can change when we open the cursor.

2. Write a PL/SQL block to display the country name and the area of each country in a chosen

region. The region\_id should be passed to the cursor as a parameter. Test your block using two

region\_ids: 5 (South America) and 30 (Eastern Asia). Do not use a cursor FOR loop.

DECLARE

 CURSOR countries\_cursor (p\_region\_id NUMBER) IS

 SELECT country\_name, area

 FROM wf\_countries

 WHERE region\_id = p\_region\_id;

 v\_countries countries\_cursor%ROWTYPE;

BEGIN

 OPEN countries\_cursor (5);

LOOP

FETCH countries\_cursor INTO v\_countries;

EXIT WHEN countries\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_countries.country\_name || ' ' || v\_countries.area);

END LOOP;

CLOSE countries\_cursor;

 OPEN countries\_cursor (30);

LOOP

FETCH countries\_cursor INTO v\_countries;

EXIT WHEN countries\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_countries.country\_name || ' ' || v\_countries.area);

END LOOP;

CLOSE countries\_cursor;

END;

3. Modify your answer to question 2 to use a cursor FOR loop. You must still declare the cursor

explicitly in the DECLARE section. Test it again using regions 5 and 30.

DECLARE

 CURSOR countries\_cursor (p\_region\_id NUMBER) IS

 SELECT country\_name, area

 FROM wf\_countries

 WHERE region\_id = p\_region\_id;

 v\_countries countries\_cursor%ROWTYPE;

BEGIN

 FOR v\_countries IN countries\_cursor (5) LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_countries.country\_name || ' ' || v\_countries.area);

END LOOP;

 FOR v\_countries IN countries\_cursor (30) LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_countries.country\_name || ' ' || v\_countries.area);

END LOOP;

END;

4. Modify your answer to question 3 to display the country\_name and area of each country in a

chosen region that has an area greater than a specific value. The region\_id and specific area

should be passed to the cursor as two parameters. Test your block twice using region\_id 5 (South

America): the first time with area = 200000 and the second time with area = 1000000.

DECLARE

 CURSOR countries\_cursor (p\_region\_id NUMBER, p\_area NUMBER) IS

 SELECT country\_name, area

 FROM wf\_countries

 WHERE region\_id = p\_region\_id AND area > p\_area;

 v\_countries countries\_cursor%ROWTYPE;

BEGIN

 FOR v\_countries IN countries\_cursor (5, 200000) LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_countries.country\_name || ' ' || v\_countries.area);

END LOOP;

 FOR v\_countries IN countries\_cursor (5, 10000000) LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_countries.country\_name || ' ' || v\_countries.area);

END LOOP;

END;

5. Modify your answer to question 4 to fetch and display two sets of countries in a single execution of the block. You should open and close the cursor twice, passing different parameter values to it

each time. Before each set of output rows, display the message “Region: <region\_id> Minimum

Area: <area>”., for example “Region: 5 Minimum Area: 200000”. Test your changes using (5,

200000) and (30, 500000).

DECLARE

CURSOR cur\_countries (p\_region\_id NUMBER, p\_area NUMBER) IS

SELECT country\_name, area

FROM wf\_countries

WHERE region\_id=p\_region\_id AND area>p\_area;

v\_country\_record cur\_countries%ROWTYPE;

BEGIN

OPEN cur\_countries(5,200000);

DBMS\_OUTPUT.PUT\_LINE('Region: ' || 5 || ' Minimum Area: ' || 200000);

LOOP

FETCH cur\_countries INTO v\_country\_record;

EXIT WHEN cur\_countries%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_country\_record.country\_name || ' with area ' || v\_country\_record.area);

END LOOP;

CLOSE cur\_countries;

DBMS\_OUTPUT.PUT\_LINE('\*\*\*\*\*\*\*\*\*\*');

OPEN cur\_countries(30,500000);

DBMS\_OUTPUT.PUT\_LINE('Region: ' || 30 || ' Minimum Area: ' || 500000);

LOOP

FETCH cur\_countries INTO v\_country\_record;

EXIT WHEN cur\_countries%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_country\_record.country\_name || ' with area ' || v\_country\_record.area);

END LOOP;

CLOSE cur\_countries;

END;