

Database Programming with PL/SQL 5-3: Cursor FOR Loops Practice Activities Vocabulary

Identify the vocabulary word for each definition below:

CURSORS FOR LOOP	Automates standard cursor-handling operations such as OPEN, FETCH, %NOTFOUND, and CLOSE so that they do not need to be coded explicitly
------------------	---

Try It / Solve It

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

Simplifies Code and Improves Readability

Improved Performance Through Automatic Row Fetching

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 countries

```

WHERE region_id = 5;
BEGIN
FOR v_countries IN countries_cur LOOP
    DBMS_OUTPUT.PUT_LINE('Country: ' || v_countries.country_name ||
        ' National holiday: ' || v_countries.national_holiday_name ||
        ', held on: ' || v_countries.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 v_countries IN (SELECT country_name, national_holiday_name, national_holiday_date
                    FROM countries
                    WHERE region_id = 5)
LOOP
    DBMS_OUTPUT.PUT_LINE('Country: ' || v_countries.country_name ||
        ' National holiday: ' || v_countries.national_holiday_name ||
        ', held on: ' || v_countries.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 v_countries IN (SELECT country_name, highest_elevation, climate
                    FROM countries
                    WHERE highest_elevation > 8000)
LOOP
    DBMS_OUTPUT.PUT_LINE('Country: ' || v_countries.country_name ||
        ' Highest Elevation: ' || v_countries.highest_elevation ||
        ' Climate: ' || v_countries.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.)

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?

5.DECLARE

```
-- Declararea cursorului pentru a prelua rile cu mai mult de 6 limbi vorbite
CURSOR country_cursor IS
  SELECT c.country_name, COUNT(sl.language) AS num_languages
  FROM COUNTRIES c
  JOIN SPOKEN_LANGUAGES sl ON c.country_id = sl.country_id
  GROUP BY c.country_name
  HAVING COUNT(sl.language) > 6;

total_countries NUMBER := 0; -- Variabil pentru numrul total de ri care îndeplinesc condiia
BEGIN
  -- Iterm prin cursor folosind un FOR loop
  FOR rec IN country_cursor LOOP
    DBMS_OUTPUT.PUT_LINE('ara: ' || rec.country_name || ' - Numr de limbi vorbite: ' || rec.num_languages);
    total_countries := total_countries + 1;
  END LOOP;

  -- Afim numrul total de ri care au mai mult de 6 limbi vorbite
  DBMS_OUTPUT.PUT_LINE('Numrul total de ri cu mai mult de 6 limbi vorbite: ' || total_countries);
END;
/
```