

PL/SQL 6-3

Agustín Alejandro Mota Hinojosa

November 26, 2023

Contents

1	Vocabulary	1
2	Try It / Solve It	1

1 Vocabulary

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

2 Try It / Solve It

1. Describe two benefits of using a cursor FOR loop.
 - (a) It's simpler
 - (b) No need to specify a counter
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 WF_COUNTRIES
        WHERE REGION_ID = 5;
BEGIN
    FOR country_rec IN countries_cur LOOP
        DBMS_OUTPUT.PUT_LINE(country_rec.COUNTRY_NAME || ' ' || country_rec.NATIONAL_HOLIDAY_NAME
                               || country_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.

```
DECLARE
    TYPE t_countries_reg IS TABLE OF WF_COUNTRIES%ROWTYPE
    INDEX BY BINARY_INTEGER;
BEGIN
    FOR rec IN (SELECT * FROM WF_COUNTRIES) LOOP
        DBMS_OUTPUT.PUT_LINE(rec.COUNTRY_NAME || ' ' || rec.NATIONAL_HOLIDAY_NAME || ' ' ||
```

```
|| 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.

```
DECLARE
```

```
TYPE t_countries_reg IS TABLE OF WF_COUNTRIES%ROWTYPE
```

```
INDEX BY BINARY_INTEGER;
```

```
BEGIN
```

```
FOR rec IN (SELECT * FROM WF_COUNTRIES WHERE HIGHEST_ELEVATION > 8000 ORDER BY HIGHEST_ELEVATION)
    DBMS_OUTPUT.PUT_LINE(rec.HIGHEST_ELEVATION || ' ' || rec.HIGHEST_ELEV_NAME || ' ' || 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_sl_count INTEGER := 6;
```

```
v_sl_total INTEGER := 0;
```

```
CURSOR countries_cur IS
```

```
SELECT c.COUNTRY_ID, c.COUNTRY_NAME, COUNT(sl.LANGUAGE_ID) AS LANGUAGE_COUNT
FROM
```

```
WF_COUNTRIES c JOIN WF_SPOKEN_LANGUAGES sl
```

```
ON c.COUNTRY_ID = sl.COUNTRY_ID
```

```
GROUP BY
```

```
c.COUNTRY_ID, c.COUNTRY_NAME
```

```
HAVING
```

```
COUNT(sl.LANGUAGE_ID) > v_sl_count;
```

```
BEGIN
```

```
FOR rec IN countries_cur LOOP
```

```
DBMS_OUTPUT.PUT_LINE(rec.country_id || ' ' || rec.COUNTRY_NAME || ' ' || rec.LANGUAGE_COUNT);
```

```
v_sl_total := v_sl_total + 1;
```

```
END LOOP;
```

```
IF v_sl_total != 0 THEN
```

```
DBMS_OUTPUT.PUT_LINE('total number of countries having more than six languages: ' || v_sl_total);
```

```
END IF;
```

```
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?

The sentence is too long.