Database Programming with PL/SQL

5-2: Using Explicit Cursor Attributes

Practice Activities **Vocabulary**

Identify the vocabulary word for each definition below:

%ROWTYPE	Declares a record with the same fields as the cursor on which it is based
Record	A composite data type in PL/SQL, consisting of a number of fields each with their own name and data type
%ISOPEN	Returns the status of the cursor
%ROWCOUNT	An attribute that processes an exact number of rows or counts the number of rows fetched in a loop
%FOUND	An attribute used to determine whether the most recent FETCH statement successfully returned a row

Try It/Solve It

1. In your own words, explain the advantage of using %ROWTYPE to declare a record structure based on a cursor declaration.

   The advantage of using %ROWTYPE to declare a record structure is that it automatically creates a record that matches the structure of the cursor's result set.

2. Write a PL/SQL block to read through rows in the countries table for all countries in region 5 (South America region). For each selected country, display the country_name, national_holiday_date, and national_holiday_name. Use a record structure to hold all the columns selected from the countries table.

   Hint: This exercise is similar to question 4G in the previous lesson. Use your solution as a starting point for this exercise.

3. For this exercise, you use the employees table. Create a PL/SQL block that fetches and displays the six employees with the highest salary. For each of these employees, display the first name, last name, job id, and salary. Order your output so that the employee with the highest salary is displayed first. Use %ROWTYPE and the explicit cursor attribute %ROWCOUNT.

4. Look again at the block you created in question 3. What if you wanted to display 21 employees instead of 6? There are only 20 rows in the employees table. What do you think would happen?

5. In real life we would not know how many rows the table contained. Modify your block from question 3 so that it will exit from the loop when either 21 rows have been fetched and displayed, or when there are no more rows to fetch. Test the block again.

1.Avantajul utilizrii %ROWTYPE pentru a declara o structur de înregistrare este c aceasta creeaz automat o înregistrare care corespunde structurii setului de rezultate al cursorului. Astfel, nu este necesar declararea manual a fiecrui câmp, ceea ce face codul mai flexibil i uor de întreinut. Dac schema tabelului se modific, codul nu trebuie actualizat manual, deoarece %ROWTYPE preia automat noile modificri.

```
2.DECLARE
    -- Declararea cursorului pentru a selecta rile din regiunea 5
    CURSOR country_cursor IS
        SELECT country_name, national_holiday_date, national_holiday_name
        FROM countries
        WHERE region_id = 5;
    -- Crearea unui registru care s corespund structurii cursorului
    country_record country_cursor%ROWTYPE;
BEGIN
    -- Deschiderea cursorului
    OPEN country_cursor;
    -- Citirea fiecrui rând i afiarea datelor
    LOOP
        FETCH country_cursor INTO country_record;
        EXIT WHEN country_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Tara: ' || country_record.country_name ||
                             ', Data sarbatorii: ' || TO_CHAR(country_record.national_holiday_date, 'DD-MON-YYYY') ||
                             ', Numele sarbatorii: ' || country_record.national_holiday_name);
    END LOOP;
    -- Închiderea cursorului
    CLOSE country_cursor;
END;
/3.DECLARE
     -- Cursor pentru a selecta primii 6 angajai cu cel mai mare salariu
     CURSOR emp_cursor IS
         SELECT first_name, last_name, job_id, salary
         FROM employees
         ORDER BY salary DESC
         FETCH FIRST 6 ROWS ONLY;
     -- Crearea unui registru bazat pe cursor
     emp_record emp_cursor%ROWTYPE;
 BEGIN
     -- Deschiderea cursorului
     OPEN emp_cursor;
     -- Citirea i afiarea fiecrui rând
     LOOP
         FETCH emp_cursor INTO emp_record;
         EXIT WHEN emp_cursor%NOTFOUND;

         DBMS_OUTPUT.PUT_LINE('Angajat: ' || emp_record.first_name || ' ' || emp_record.last_name ||
                              ', Job ID: ' || emp_record.job_id ||
                              ', Salariu: ' || emp_record.salary);
     END LOOP;
     -- Închiderea cursorului
     CLOSE emp_cursor;
 END;
```

4.În cazul în care am modifica interogarea pentru a extrage 21 de angajai, dar tabelul employees conine doar 20 de rânduri, atunci cursorul ar încerca s extrag un rând suplimentar care nu exist.

În acest caz, la a 21-a încercare de preluare (FETCH), cursorul nu va gsi date i va seta implicit %NOTFOUND pe TRUE, ceea ce poate fi folosit pentru a iei corect din bucl.

Dac nu verificm aceast condiie, programul poate intra într-o stare nedorit sau s încerce s acceseze date inexistente.

5.DECLARE
```
    -- Cursor pentru a selecta angajaii cu cele mai mari salarii
    CURSOR emp_cursor IS
        SELECT first_name, last_name, job_id, salary
        FROM employees
        ORDER BY salary DESC;

    -- Crearea unui registru bazat pe cursor
    emp_record emp_cursor%ROWTYPE;

    -- Contor pentru numrarea rândurilor afiate
    row_count NUMBER := 0;

BEGIN
    -- Deschiderea cursorului
    OPEN emp_cursor;

    -- Citirea i afiarea fiecrui rând, cu verificare pentru maxim 21 de rânduri
    LOOP
        FETCH emp_cursor INTO emp_record;
        EXIT WHEN emp_cursor%NOTFOUND OR row_count = 21;

        -- Incrementarea contorului
        row_count := row_count + 1;

        -- Afiarea informaiilor angajatului
        DBMS_OUTPUT.PUT_LINE('Angajat: ' || emp_record.first_name || ' ' || emp_record.last_name ||
                             ', Job ID: ' || emp_record.job_id ||
                             ', Salariu: ' || emp_record.salary);
    END LOOP;

    -- Închiderea cursorului
    CLOSE emp_cursor;
END;
/
```

Explicaia modificrilor:

Am adugat o variabil row_count pentru a ine evidena numrului de angajai afiai.

În bucla LOOP, verificm atât dac %NOTFOUND este TRUE, cât i dac row_count a ajuns la 21.

Dac oricare dintre aceste condiii este adevrat, ieim din bucl (EXIT).

Aceast metod asigur c nu încercm s extragem mai multe rânduri decât exist în tabel.