Database Programming with PL/SQL

6-2: Using Explicit Cursor Attributes Practice Activities

Vocabulary

Identify the vocabulary word for each definition below:

%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
%NOTFOUND	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.

   Se evita el tener que declarar variables, solo basta con declarar un registro y se ahorra más líneas de código.

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.

   ```
   DECLARE
   cursor cur_countries IS SELECT country_name, NATIONAL_HOLIDAY_DATE,
   NATIONAL_HOLIDAY_NAME
   FROM countries WHERE NATIONAL_HOLIDAY_DATE IS NOT NULL AND REGION_ID =5;
   v_countries cur_countries%ROWTYPE;
   ```

```
BEGIN
open cur_countries;
LOOP
FETCH cur_countries INTO v_countries;
EXIT WHEN cur_countries%NOTFOUND;
DBMS_OUTPUT.PUT_LINE(v_countries.country_name || ' '||
v_countries.NATIONAL_HOLIDAY_DATE || ' '|| v_countries.NATIONAL_HOLIDAY_NAME );
END LOOP;
CLOSE cur_countries;
END;
```


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.

```
DECLARE
cursor cur_employees IS select first_name, last_name, job_id,salary from employees
order by salary desc;
rec_employees cur_employees%ROWTYPE;
BEGIN
open cur_employees;
LOOP
FETCH cur_employees INTO rec_employees;
EXIT WHEN cur_employees%NOTFOUND OR cur_employees%rowcount > 6;
DBMS_OUTPUT.PUT_LINE(rec_employees.first_name||' '||rec_employees.last_name||'
'||rec_employees.job_id||' '|| rec_employees.salary);
END LOOP;
CLOSE cur_employees;
END;
```

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?

Mostrará todas las filas y se detendrá cuando no encuentre otra fila más.

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.

```
DECLARE
cursor cur_employees IS select first_name, last_name, job_id,salary from employees order by salary desc;
rec_employees cur_employees%ROWTYPE;
BEGIN
open cur_employees;
LOOP
FETCH cur_employees INTO rec_employees;
EXIT WHEN cur_employees%NOTFOUND OR cur_employees%rowcount > 21;
DBMS_OUTPUT.PUT_LINE(rec_employees.first_name||'
'||rec_employees.last_name||' '||rec_employees.job_id||' '|| rec_employees.salary);
END LOOP;
CLOSE cur_employees;
END;
```