



LABORATOR 4

Instruciuni iterative. Cursoare



Instructiuni iterative - exercitii

(mai multe informatii si sintaxa in lab. 3)

1. Folosind o instructiune *LOOP* calculați $n!$.
2. Sa se declare si sa se initializeze cu 1 variabila *v_i* de tip *POSITIVE* si cu 10 constanta *c_max* de tip *POSITIVE*. Sa se implementeze un ciclu *LOOP* care incrementeaza pe *v_i* pâna când acesta ajunge la o valoare $> c_max$, moment în care ciclul *LOOP* este parasit si se sare la instructiunea *v_i:=1*. (*GO TO/EXIT*). Se cer 2 metode.
3. Creati structura tabelului *org_tab* constând din trei coloane, *cod_tab* de tip *INTEGER* ce contine un contor al înregistrărilor, *cod_nr* de tip *integer* ce contine nr. multiplu de 100 si *text_tab* de tip *VARCHAR2* ce contine un text asociat fiecărei înregistrări- par/impar in functie de paritatea coloanei *cod_tab*. Sa se introduca 70 de înregistrări în acest tabel. Se cer 2 metode.
4. Să se scrie un bloc *PL/SQL* care face o listă cu numărul angajaților care au veniturile mai mici de 4000 (considerate venituri mici) și mai mari de 4000 (considerate venituri mari), pentru fiecare departament. Se va face o construcție folosind structuri de control și cicluri. Pentru departamentele care nu au niciun angajat se va afișa un mesaj.

Cursoare

- Pentru executia cererilor SQL prezente in blocurile PL/SQL si pentru stocarea rezultatelor acestora (cum e cazul cererilor SELECT) Oracle foloseste un spatiu de lucru accesibil utilizatorului prin intermediul unui obiect numit cursor.
- Un cursor este o modalitate de a parcurge (linie cu linie) multimea de linii procesate returnate de o cerere 'multiple-row'. Aceasta multime se numeste *active set*.
- Tipuri:
 - *Cursoare implicite*
 - *Cursoare explicite*

Cursoare implicite

- PL/SQL declara un cursor implicit pentru fiecare operatie de manipulare a datelor in SQL, incluzand interogari care returneaza o singura inregistrare.
- Cursoarele implicite sunt administrate automat de PL/SQL, deci nu trebuie scris nici un cod pentru manipularea lor. Cursorul implicit poate fi manipulat prin intermediul atributelor lui.
- Atributele cursoarelor pot returna informatii despre executia comenzilor DML si DDL, cum ar fi INSERT, UPDATE, DELETE, SELECT INTO, COMMIT sau ROLLBACK. Atributele cursorului sunt: %FOUND, %ISOPEN, %NOTFOUND si %ROWCOUNT.

- %FOUND- Pana cand comanda SQL de manipulare a datelor este executata, atributul %FOUND este NULL. Deci, %FOUND este TRUE daca o comanda de tipul INSERT, UPDATE sau DELETE afecteaza una sau mai mult inregistrari din baza de date sau SELECT INTO returneaza una sau multe inregistrari
- %ISOPEN- Oracle inchide cursorul automat dupa executarea comenzilor. Ca rezultat, %ISOPEN devine FALSE
- %NOTFOUND- %NOTFOUND este atributul opus lui %FOUND. %NOTFOUND este TRUE daca INSERT, UPDATE sau DELETE nu afecteaza nici o inregistrare din baza de date sau SELECT INTO nu returneaza nicio inregistrare. Altfel este FALSE.
- %ROWCOUNT- %ROWCOUNT returneaza numarul de inregistrari afectate de una din comenzile INSERT, UPDATE, sau DELETE sau numarul de inregistrari afectate de SELECT INTO. %ROWCOUNT este 0 daca comanda INSERT, UPDATE, sau DELETE nu afecteaza nici o inregistrare, sau SELECT INTO nu returneaza nicio inregistrare.

Cursoare explicite:

- pentru cereri care returneaza mai mult de o linie, sunt definite cursoare explicite, denumite de programator si manipulate prin intermediul unor comenzi specifice.

➤ **Etapele** utilizarii unui cursor:

a) **Declarare** (în secțiunea declarativa a blocului PL/SQL):

CURSOR *c_nume_cursor* [(*parametru tip_de_Date, ..*)] *IS*
Comanda SELECT;

b) **Deschidere** (comanda OPEN), operație ce identifică mulțimea de linii (*active set*):

OPEN *c_nume_cursor* [(*parametru, ...*)];

c) **Incarcare** (comanda FETCH). Numarul de variabile din clauza INTO trebuie sa se potriveasca cu lista SELECT returnata de cursor.

FETCH *c_nume_cursor* **INTO** *variabila, ...;*

d) **Verificare** dacă nu am ajuns cumva la finalul mulțimii de linii folosind atributele:

C_nume_cursor%NOTFOUND – valoare booleana

C_nume_cursor%FOUND – valoare booleana

Daca nu s-a ajuns la final mergi la c).

e) **Inchidere** cursor (operațiune foarte importanta avand in vedere ca daca nu e inchis cursorul ramane deschis si consuma din resursele serverului, MAX_OPEN_CURSORS)

CLOSE *c_nume_cursor;*

Valorile atributelor unui cursor explicit

	OPEN		Primul FETCH		Următorul FETCH		Ultimul FETCH		CLOSE	
	Înainte	După	Înainte	După	Înainte	După	Înainte	După	Înainte	După
%ISOPEN	False	True	True	True	True	True	True	True	True	False
%FOUND	Eroare	Null	Null	True	True	True	True	False	False	Eroare
%NOTFOUND	Eroare	Null	Null	False	False	False	False	True	True	Eroare
%ROWCOUNT	Eroare	0	0	1	1	Depinde de date				Eroare

➤ Clauza FOR UPDATE

Comanda SELECT are urmatoarea extensie PL/SQL pentru blocarea explicita inregistrarilor ce urmeaza a fi prelucrate (modificate sau sterse):

```
SELECT ...  
FROM ...  
WHERE ...  
  
...  
ORDER BY ...  
FOR UPDATE [OF lista_coloane] [NOWAIT | WAIT n];
```

- Daca liniile selectate de cerere nu pot fi blocate din cauza altor blocari atunci
 - daca se foloseste NOWAIT este ridicata imediat eroarea ORA-00054
 - daca nu se foloseste NOWAIT atunci se asteapta pana cand liniile sunt deblocate.
 - daca se foloseste WAIT n atunci se asteapta un numar determinat de secunde inainte de a da eroare ca liniile ce trebuie selectate pentru modificare sunt blocate.
- Nu este recomandata anularea (ROLLBACK) sau permanentizarea schimbarilor inainte de a inchide cursorul ce foloseste FOR UPDATE pentru ca aceasta ar elibera blocarile realizate de acesta.
- Pentru a modifica o anumită linie returnata de un cursor se poate folosi clauza:

WHERE CURRENT OF nume_cursor

Aceasta clauza apare la finalul unei comenzi UPDATE si face referinta la un cursor care este deschis si s-a facut cel putin o incarcare din el (FETCH).

Exercitii

1. Ce elemente lipsesc din urmatorul bloc?

```
DECLARE
    CURSOR emp_cur IS SELECT * FROM emp;
BEGIN
    OPEN emp_cur;
    FETCH emp_cur INTO emp_rec;
END;
```

2. Ce element ar trebui eliminat din urmatorul bloc?

```
DECLARE
    CURSOR emp_cur IS SELECT * FROM emp;
    emp_rec emp_cur%ROWTYPE;
BEGIN
    FOR emp_rec IN emp_cur
    LOOP
        give_raise (emp_rec.empno);
    END LOOP;
END;
```

3. Ce se afiseaza? La ce comanda se refera SQL%?

```
BEGIN
    DELETE FROM tip_plata
    WHERE id_tip_plata NOT IN (
        SELECT id_tip_plata
        FROM facturi);
    --cursor deschis?
    IF SQL%ISOPEN THEN DBMS_OUTPUT.PUT_LINE('Cursor deschis');
    ELSE DBMS_OUTPUT.PUT_LINE('Cursor inchis');
    END IF;
    --SQL% se refera la comanda DELETE sau la comanda SELECT ?
END;
```

/

3. Pentru fiecare din urmatoarele afirmatii, numiti atributul (impreuna cu numele cursorului) care ar trebui utilizat:
- a) *Daca "FETCH" nu intoarce nicio inregistrare din cursorul "company_cursor", paraseste "LOOP".*
 - b) *Daca numarul de linii sterse depaseste 100, notificati managerul.*
 - c) *Daca "emp_cursor" este deja deschis, obtine urmatoarea inregistrare. Altfel, deschide cursorul.*
 - d) *Daca "FETCH" intoarce (cel putin) o linie din cursorul "sales_cursor", afisati informatii despre totalul vanzarilor.*
 - e) *Daca se utilizeaza un cursor implicit intr-o clauza "SELECT" pentru a obtine informatii despre ultima comanda dintr-un magazine specificat, iar aceasta interogare nu intoarce nicio linie, afisati un mesaj de avertizare.*

4. Ce mesaj se va afisa daca "SELECT"-ul nu intoarce nicio linie?

```
DECLARE
```

```
    emp_id  INTEGER;
```

```
    department# departments.department_id%TYPE := NULL;
```

```
BEGIN
```

```
    SELECT department_id
```

```
    INTO department#
```

```
    FROM employees
```

```
    WHERE employee_id = emp_id;
```

```
    IF department# IS NULL THEN
```

```
        dbms_output.put_line('Dept is not found!');
```

```
    ELSE
```

```
        dbms_output.put_line('Dept is ' || to_char(department#));
```

```
    END IF;
```

```
    EXCEPTION
```

```
        WHEN no_data_found THEN dbms_output.put_line('No data found');
```

```
    END;
```

```
/
```

5. Cum ar trebui modificat urmatorul bloc pentru a se afisa suma salariilor din fiecare department?

```
DECLARE
    CURSOR tot_cursor IS
        SELECT department_id, SUM (salary)
        FROM emp
        GROUP BY department_id;
BEGIN
    FOR tot_rec IN tot_cursor LOOP
        DBMS_OUTPUT.PUT_LINE ('Total is: ' || tot_rec.total_sales);
    END LOOP;
END;
/
```

6. Să se obțină câte o linie de forma ' <nume> are salariul anual <salariu anual> pentru fiecare angajat din departamentul 50. Se cer 4 soluții (WHILE, LOOP, FOR specific cursorilor, FOR cu varianta de scriere a cursorului în interiorul său).
7. Să se actualizeze liniile tabelului emp, mărin­d cu 10% valoarea comisionului pentru salariații având salariul mai mic decât o valoare introdusă de utilizator. Să se afișeze dacă au fost actualizate linii sau nu (SQL%FOUND), iar în caz afirmativ să se afișeze numărul de linii afectate (SQL%ROWCOUNT). Ce fel de cursor folosim?

8. Rescrieti urmatorul bloc astfel incat id-ul departamentului sa fie introdus de utilizator si id-ul departamentului sa fie dat ca parametru cursorului.

```
DECLARE
    CURSOR dept10_cur IS
        SELECT dname, SUM (sal) total_sales
        FROM emp
        WHERE deptno = 10;
    dept10_rec dept10_cur%ROWTYPE;
    CURSOR dept20_cur IS
        SELECT dname, SUM (sal)
        FROM emp
        WHERE deptno = 20;
    dept20_rec dept20_cur%ROWTYPE;
BEGIN
    OPEN dept10_cur;
    FETCH dept10_cur INTO dept10_rec;
    DBMS_OUTPUT.PUT_LINE
        ('Total for department 10 is: ' || tot_rec.total_sales);
    CLOSE dept10_cur;
    OPEN dept20_cur;
    FETCH dept20_cur INTO dept20_rec;
    DBMS_OUTPUT.PUT_LINE
        ('Total for department 20 is: ' || tot_rec.total_sales);
    CLOSE dept20_cur;
END;
```

9. Creați un bloc PL/SQL care determină cele mai mari n salarii, urmând pașii descriși în continuare:
- a)creați un tabel top_salarii_pnu, având o coloană salary.
 - b)Numărul n (al celor mai bine plătiți salariați) se va introduce de către utilizator (se va folosi o variabilă de substituție p_num).
 - c)În secțiunea declarativă a blocului PL/SQL se vor declara 2 variabile: v_num de tip NUMBER (corespunzătoare lui p_num) și v_sal de tipul coloanei salary. Se va declara un cursor emp_cursor pentru regăsirea salariilor în ordine descrescătoare (se presupune că nu avem valori duplicate).
 - d)Se vor introduce cele mai mari mai bine plătiți n angajați în tabelul top_salarii_pnu;
 - e)Afișați conținutul tabelului top_salarii_pnu.
 - f)Testați cazuri speciale, de genul $n = 0$ sau n mai mare decât numărul de angajați. Se vor elimina înregistrările din tabelul top_salarii_pnu după fiecare test.
10. Obțineți pentru fiecare departament numele acestuia și numărul de angajați, într-una din următoarele forme:
- “ În departamentul <nume departament> nu lucrează angajati”.
 - “ În departamentul <nume departament> lucrează un angajat”.
 - “ În departamentul <nume departament> lucrează <numar> angajati”.
11. Modificați exercițiul 10 astfel încât să obțineți doar departamentele în care lucrează cel puțin x angajați, unde x reprezintă un număr introdus de la tastatură. Rezolvați problema folosind toate tipurile de cursoare studiate.

11. Modificați exercițiul 10 astfel încât să obțineți doar departamentele în care lucrează cel puțin x angajați, unde x reprezintă un număr introdus de la tastatură. Rezolvați problema folosind toate tipurile de cursoare studiate.
12. Obțineți primii 3 manageri care au cei mai mulți subordonați. Afișați numele managerului, respectiv numărul de angajați.
 - a. Rezolvați problema folosind un cursor explicit.
 - b. Modificați rezolvarea anterioară astfel încât să obțineți primii 4 manageri care îndeplinesc condiția. Observați rezultatul obținut și specificați dacă la punctul a s-a obținut top 3 manageri?
13. Sa se dubleze valoarea salariilor angajatilor înainte de 1 ianuarie 1995, care nu câștiga comision.
(cursor for update)
14. Să se declare un cursor cu un parametru de tipul coloanei `location_id`, care determină departamentele din locația respectivă și blochează liniile pe perioada prelucrării acestora. Să se deschidă cursorul folosind o variabilă de substituție pentru furnizarea parametrului. Să se actualizeze tabelul `dep_pnu`, dând valoarea 100 locației corespunzătoare liniei curente a cursorului.
15. Modificati exemplul de mai sus astfel incat noua valoare a numelui departamentelor afectate de bloc sa fie vechea valoare la care se adauga numele locatiei date ca parametru.