SGBD-Anul 3
Laborator 3

Cursoare

Un cursor este o modalitate de a parcurge (linie cu linie) multimea de linii procesate returnate de o cerere 'multiple-row'. Această mulțime se numeşte *active set*.

- Cursoarele pot fi:
  - implicite – care sunt declarate de PL/SQL in mod implicit pentru toate comenzile LMD si comanda SELECT, inclusiv comenzile care returneaza o singura linie.
  - 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 sectiunea declarativa a blocului PL/SQL):
CURSOR *c_nume_cursor* [*(parametru tip_de_Date, ..)*] *IS*
Comanda SELECT;

b) ***Deschidere*** (comanda OPEN), operatie 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 (operatiune 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;*

- **Atributele** cursoarelor

Atribut	**Tip**	**Descriere**
%ISOPEN	Boolean	TRUE atunci când cursorul este deschis
%NOTFOUND	Boolean	TRUE dacă cea mai recentă operație FETCH nu a regăsit o linie
%FOUND	Boolean	TRUE dacă cea mai recentă operație FETCH a întors o linie
%ROWCOUNT	Number	Întoarce numărul de linii returnate până la momentul respectiv.

- **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).

Exerciții: [Cursoare implicite]

1. Să se actualizeze liniile tabelului emp_pnu, mărind cu 10% valoarea comisionului pentru salariații având salariul mai mic decât o valoare introdusă de utilizator. Să se afişeze dacă au fost actualizate linii sau nu (SQL%FOUND), iar în caz afirmativ să se afişeze numărul de linii afectate (SQL%ROWCOUNT). Ce fel de cursor folosim?

2. Să se creeze un tabel DEP_EMP_PNU având câmpurile cod_dep şi cod_ang. Să se introducă într-o variabilă de tip tablou imbricat codurile departamentelor (în care există angajați), iar apoi, prin intermediul unei comenzi FORALL să se insereze aceste coduri şi codurile angajaților corespunzători în tabelul DEP_EMP_PNU. Pentru fiecare departamentm să se afişeze câți angajați au fost introduşi (SQL%BULK_ROWCOUNT(iteratie)).

Exerciții: [Introducere cursoare explicite]

3. Să se obțină câte o linie de forma ‘ <nume> are salariul anual <salariu annual> pentru fiecare angajat din departamentul 50. Se cer 4 soluții (WHILE, LOOP, FOR specific cursoarelor, FOR cu varianta de scriere a cursorului în interiorul său).

4. Sa se afiseze salariatii care au salariul mai mic de 7000$, in urmatoarea forma:

-Salariatul <nume> castiga <salariu>-.

5. Creați un bloc PL/SQL care determină cele mai mari n salarii, urmând paşii descrişi 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 varianilă 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) şi 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) Afişați conținurtul 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.

Exerciții: [Cursoare cu parametru]

6. Să se declare un cursor cu un parametru de tipul codului angajatului, care regăseşte numele şi salariul angajaților având codul transmis ca parametru sau numele si salariile tuturor angajatilor daca valoarea parametrului este null. Să se declare o variabilă v_nume de tipul unei linii a cursorului. Să se declare două tablouri de nume (v_tab_nume), respectiv salarii (v_tab_sal). Să se parcurgă liniile cursorului în două moduri: regăsindu-le în v_nume sau în cele două variabile de tip tablou.

7. Utilizând un cursor parametrizat să se obțină codurile angajaților din fiecare departament şi pentru fiecare job. Rezultatele să fie inserate în tabelul *mesaje*, sub forma câte unui şir de caractere obținut prin concatenarea valorilor celor 3 coloane.

Exerciții: [FOR UPDATE, WHERE CURRENT OF]

8. Să se dubleze valoarea salariilor angajaților înainte de 1 ianuarie 1995, care nu câştigă comision (testat cu WAIT si NOWAIT).

9. Să se declare un cursor cu un parametru de tipul coloanei location_id, care determină departamentele din locația respectivă şi 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.

10. 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.

Exerciții [Cursoare dinamice]

11. Să se declare un cursor dinamic care întoarce linii de tipul celor din tabelul emp_pnu. Să se citească o opțiune de la utilizator, care va putea lua valorile 1, 2 sau 3. Pentru opțiunea 1 deschideți cursorul astfel încât să regăsească toate informațiile din tabelul EMP_pnu, pentru opțiunea 2, cursorul va regăsi doar angajații având salariul cuprins între 10000 şi 20000, iar pentru opțiunea 3 se vor regăsi salariații angajați în anul 1990.

12. Să se citească o valoare n de la tastatura. Prin intermediul unui cursor deschis cu ajutorul unui şir dinamic să se regăsească angajații având salariul mai mare decât n. Pentru fiecare linie regăsită de cursor, dacă angajatul are comision, să se afişeze numele său şi salariul.

Exerciții: [Expresii cursor]

13. Să se listeze numele regiunilor şi pentru fiecare regiune să se afişeze numele țărilor. Se cer 2 metode de rezolvare (secvențial şi cu expresii cursor).

Exerciții propuse:

1. Sa se afiseze departamentele si media salariilor pe departamente, in urmatoarea forma:
   - In departamentul <nume departament> media salariilor este <media>.

2. Sa se afiseze primii 5 salariati considerati in ordine alfabetica.

3. Să se determine primii n cel mai bine plătiți salariați, în ipoteza că avem valori duplicat.
Exemplu : Dacă primii angajati au salariile
A 1000
B 700
C 700
D 800
Iar utilizatorul introduce n=2, se vor regăsi
A 1000
B 700
C 700

4. Scrieti o cerere pentru a regasi toate job-urile si salariatii practicând fiecare job. Rezultatele se vor scrie in tabelul *MESAJE*. Se va folosi un cursor pentru a regasi codul job-ului si se va transmite acest cod unui cursor care regaseste salariatii având job-ul respectiv.

5. Să se scrie un bloc PL/SQL care declară şi utilizează un cursor cu parametri, astfel:într-o instrucțiune de ciclare, utilizați un cursor care regăseşte codul şi numele departamentelor având codul mai mic decât 100. Transmiteți ca parametru codul departamentului către alt cursor pentru regăsirea, în tabelul employees, a numelui job-ului, datei angajării şi salariului angajaților având codul mai mic decât 20 şi care lucrează în acel departament.

6. a) Utilizatorul va putea specifica un cod de departament prin intermediul unei variabile de substituție p_dep.
b) Să se creeze un bloc PL/SQL care declară o variabilă corespunzătoare variabilei de substituție şi un cursor, emp_cursor, pentru regăsirea numelui, salariului şi codului şefului pentru angajații din departamentul specificat în p_dep.
Utilizați cursorul pentru a realiza următoarele:
   - dacă salariul angajatului este mai mic decât 5000 şi codul şefului său este 101 sau 124, să se afişeze '<nume> va fi propus pentru marire';
   - altfel, se va afişa '<nume> nu va fi propus pentru mărire'.

7. a) Creați un cursor care regăseşte salariul, prenumele şi numele angajaților dintr-un departament al cărui cod este introdus de utilizator. Utilizați clauza FOR UPDATE. Ce efect are aceasta?
b) Se presupune că doar angajații din departamentele 20, 60, 80, sau 100 pot fi propuşi pentru o mărire de salariu. Să se verifice dacă utilizatorul a introdus unul dintre aceste departamente. In caz afirmativ, se va deschide cursorul, altfel se va afişa un mesaj corespunzător.
c) utilizând o expresie CASE , să se stabilească următoarele:
   - salariile <6500 vor fi mărite cu 20% ;
   - salariile >6500 şi <9500 vor fi mărite cu 15%;
   - salariile >9500 şi <12000 vor fi mărite cu 8%;
   - salariile >12000 vor fi mărite cu 3%.