

Baze de date-Anul 3 (IDD)

Laborator 3 PL/SQL

Cursoare

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

➤ Cursoarele pot fi:

- implicite – care sunt declarate de PL/SQL în mod implicit pentru toate comenzile LMD și comanda SELECT, inclusiv comenzile care returnează o singură linie.
- explicite – pentru cereri care returnează mai mult de o linie, sunt definite cursoare explicite, denumite de programator și manipulate prin intermediul unor comenzi specifice.

➤ **Etapele** utilizării unui cursor:

a) **Declaraire** (în secțiunea declarativă 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). Numărul de variabile din clauza INTO trebuie să se potrivească cu lista SELECT returnată 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 booleană

C_nume_cursor%FOUND – valoare booleană

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

e) **Inchidere** cursor (operațiune foarte importantă având în vedere că dacă nu e închis cursorul rămâne deschis și consumă 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.

- Pentru a modifica o anumită linie returnată de un cursor se poate folosi clauza:

WHERE CURRENT OF *nume_cursor*

Această clauză apare la finalul unei comenzi UPDATE și face referința la un cursor care este deschis și s-a făcut cel puțin o încărcare din el (FETCH).

Exerciții: [Introducere cursoare explicite]

1. 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 cursoarelor cu varianta de scriere a cursorului în interiorul său).

Soluția 1:

DECLARE

```

CURSOR c_emp IS
    SELECT last_name, salary*12 sal_an
    FROM emp_pnu
    WHERE department_id = 50;
V_emp c_emp%ROWTYPE;
BEGIN
    OPEN c_emp;
    FETCH c_emp INTO v_emp;
    WHILE (c_emp%FOUND) LOOP
        DBMS_OUTPUT.PUT_LINE (' Nume:' || v_emp.last_name ||
            ' are salariul anual : ' || v_emp.sal_an);
        FETCH c_emp INTO v_emp;
    END LOOP;
    CLOSE c_emp;
END;

```

Soluția 2:

```

DECLARE
    CURSOR c_emp IS
        SELECT last_name, salary*12 sal_an
        FROM emp_pnu
        WHERE department_id = 50;
V_emp c_emp%TYPE;
BEGIN
    OPEN c_emp;
    LOOP
        FETCH c_emp INTO v_emp;
        EXIT WHEN c_emp%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE (' Nume:' || v_emp.last_name ||
            ' are salariul anual : ' || v_emp.sal_an);
    END LOOP;
    CLOSE c_emp;
END;

```

Soluția 3: // nu mai este nevoie explicit de OPEN, FETCH, CLOSE !!!

```

DECLARE
    CURSOR c_emp IS
        SELECT last_name, salary*12 sal_an
        FROM emp_pnu
        WHERE department_id = 50;
BEGIN
    FOR v_emp IN c_emp LOOP
        DBMS_OUTPUT.PUT_LINE (' Nume:' || v_emp.last_name ||
            ' are salariul anual : ' || v_emp.sal_an);
    END LOOP;
END;

```

Soluția 4:

```

BEGIN
    FOR v_rec IN (SELECT last_name, salary*12 sal_an
        FROM employees
        WHERE department_id = 50) LOOP
        DBMS_OUTPUT.PUT_LINE (' Nume:' || v_rec.last_name ||
            ' are salariul anual : ' || v_rec.sal_an);
    END LOOP;
END;
/

```

2. Creați un bloc PL/SQL care determină cele mai mari n salarii, urmând pașii descriși în continuare:
- creați un tabel `top_salarii_pnu`, având o coloană `salary`.
 - Numărul n (al celor mai bine plătiți salariați) se va introduce de către utilizator (se va folosi comanda SQL*Plus ACCEPT și o variabilă de substituție `p_num`).
 - 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*).
 - Se vor introduce cele mai mari n mai bine plătiți angajați în tabelul `top_salarii_pnu`;
 - Afișați conținutul tabelului `top_salarii_pnu`.
 - 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.

Soluție:

```
ACCEPT p_num PROMPT ' ... '
DECLARE
    V_num  NUMBER(3) := &p_num;
    V_sal   emp_pnu.salary%TYPE;
    CURSOR emp_cursor IS
        SELECT DISTINCT salary
        FROM   emp_pnu
        ORDER BY salary DESC;
BEGIN
    OPEN emp_cursor; -- folosit si alte variante de lucru cu cursorul !!!!
    FETCH emp_cursor INTO v_sal;
    WHILE emp_cursor%ROWCOUNT <= v_num AND emp_cursor%FOUND LOOP
        INSERT INTO top_salarii_pnu (salary)
            VALUES (v_sal);
        FETCH emp_cursor INTO v_sal;
    END LOOP;
    CLOSE emp_cursor;
END;
/
SELECT * FROM top_salarii_pnu;
```

Exerciții: [Cursoare cu parametru]

3. Să se declare un cursor cu un parametru de tipul codului angajatului, care regăsește numele și salariul angajaților având codul transmis ca parametru sau ale numele și salariile tuturor angajaților dacă valoarea parametrului este null. Să se declare o variabilă `v_ume` de tipul unei linii a cursorului. Să se declare două tablouri de nume (`v_tab_ume`), respectiv salarii (`v_tab_sal`). Să se parcurgă liniile cursorului în două moduri: regăsindu-le în `v_ume` sau în cele două variabile de tip tablou.

```
DECLARE
    CURSOR c_ume (p_id employees.employee_id%TYPE) IS
        SELECT last_name, salary
        FROM   employees
        WHERE  p_id IS NULL // conditia este adevarata pentru toate liniile tabelului atunci cand
                          // parametrul este null
        OR    employee_id = p_id;
    V_ume c_ume%ROWTYPE;
    -- sau
    /* TYPE t_ume IS RECORD
        (last_name employees.employee_id%TYPE,
         salary   employees.salary%TYPE
        v_ume t_ume;
    */
```

```

TYPE t_tab_nume IS TABLE OF employees.last_name%TYPE;
TYPE t_tab_sal IS TABLE OF employees.salary%TYPE;
V_tab_nume t_tab_nume;
V_tab_sal t_tab_sal;
BEGIN
  IF c_nume%ISOPEN THEN
    CLOSE c_nume;
  END IF;
  OPEN c_nume (104);
  FETCH c_nume INTO v_nume;
  WHILE c_nume%FOUND LOOP
    DBMS_OUTPUT.PUT_LINE (' Nume:' || v_nume.last_name ||
                          ' salariu : ' || v_nume.salary);
    FETCH c_nume INTO v_nume;
  END LOOP;
  CLOSE c_nume;
  -- eroare INVALID CURSOR
  -- FETCH c_nume INTO v_nume;
  DBMS_OUTPUT.PUT_LINE (' SI o varianta mai eficienta');
  OPEN c_nume (null);
  FETCH c_nume BULK COLLECT INTO v_tab_nume, v_tab_sal;
  CLOSE c_nume;
  FOR i IN v_tab_nume.FIRST..v_tab_nume.LAST LOOP
    DBMS_OUTPUT.PUT_LINE (i || ' Nume:' || v_tab_nume(i) ||
                          ' salariu : ' || v_tab_sal(i));
  END LOOP;
END;
/

```

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

```

DECLARE
  v_cod_dep departments.department_id%TYPE;
  v_cod_job departments.job_id%TYPE;
  v_mesaj VARCHAR2(75);
  CURSOR dep_job IS
    SELECT department_id, job_id
    FROM emp_pnu;
  CURSOR emp_cursor (v_id_dep NUMBER, v_id_job VARCHAR2) IS
    SELECT employee_id || department_id || job_id
    FROM emp_pnu
    WHERE department_id = v_id_dep
    AND job_id = v_id_job;
BEGIN
  OPEN dep_job;
  LOOP
    FETCH dep_job INTO v_cod_dep, v_cod_job;
    EXIT WHEN dep_job%NOTFOUND;
    IF emp_cursor%ISOPEN THEN
      CLOSE emp_cursor;
    END IF;
    OPEN emp_cursor (v_cod_dep, v_cod_job);
    LOOP
      FETCH emp_cursor INTO v_mesaj;
      EXIT WHEN emp_cursor%NOTFOUND;
      INSERT INTO mesaje (rezultat)

```

```

VALUES (v_mesaj);
END LOOP;
CLOSE emp_cursor;
END LOOP;
CLOSE dep_job;
COMMIT;
END;
/

```

Exerciții [Cursoare dinamice]

5. 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 și 20000, iar pentru opțiunea 3 se vor regăsi salariații angajați în anul 1990.

```

ACCEPT p_optiune PROMPT 'Introduceti optiunea (1,2 sau 3) '
DECLARE
  TYPE emp_tip IS REF CURSOR RETURN emp_pnu%ROWTYPE;
  V_emp emp_tip;
  V_optiune NUMBER := &p_optiune;
BEGIN
  IF v_optiune = 1 THEN
    OPEN v_emp FOR SELECT * FROM emp_pnu;
    --!!! Introduceți cod pentru afișare
  ELSIF v_optiune = 2 THEN
    OPEN v_emp FOR SELECT * FROM emp_pnu
      WHERE salary BETWEEN 10000 AND 20000;
    --!!! Introduceți cod pentru afișare
  ELSIF v_optiune = 3 THEN
    OPEN emp_pnu FOR SELECT * FROM emp_pnu
      WHERE TO_CHAR(hire_date, 'YYYY') = 1990;
    --!!! Introduceți cod pentru afișare
  ELSE
    DBMS_OUTPUT.PUT_LINE('Optiune incorecta');
  END IF;
END;
/

```

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

```

DECLARE
  TYPE empref IS REF CURSOR;
  V_emp empref;
  v_nr INTEGER := &p_nr;
BEGIN
  OPEN v_emp FOR
    'SELECT employee_id, salary FROM emp_pnu WHERE salary > :bind_var'
    USING v_nr;
  -- introduceți liniile corespunzătoare rezolvării problemei
END;

```