

Tipul de obiect CURSOR

Obiectele de tip cursor reprezintă mecanismul prin care limbajul PL/SQL permite accesarea și manipularea rândurilor returnate de o instrucțiune SELECT. Deoarece o comandă SQL poate returna mai multe înregistrări, un cursor oferă o modalitate controlată de a parcurge aceste rezultate, rând cu rând, în cadrul unui bloc PL/SQL.

Un cursor poate fi privit ca un „indicator” sau un „mâner” către un set de rezultate generat de o interogare. Prin intermediul său, programatorul poate deschide interogarea, citi valorile rândurilor succesive și, în final, închide cursorul pentru a elibera resursele ocupate. De asemenea, cursoarele pot fi utilizate pentru a actualiza sau șterge rânduri din tabelele sursă, dacă sunt declarate corespunzător.

Tipuri de cursoare

Limbajul PL/SQL ne pune la dispoziție diverse moduri de a utiliza cereri SQL în cadrul programelor, iar funcționalitatea aceasta extinsă se datorează în mare parte tipurilor de cursoare utilizate. Orice cerere SQL integrată într-un bloc de cod este realizată printr-un cursor, fie acesta declarat explicit sau nu.

Cursoare implicate

Acestea sunt gestionate automat de către PL/SQL pentru fiecare instrucțiune INSERT, UPDATE, DELETE sau SELECT INTO. Programatorul nu le declară explicit, nefiind necesară nici închiderea lor pentru a elibera memoria. Se pot accesa doar atributele lor prin prefixul SQL%, de exemplu:

```
IF SQL%FOUND THEN
    DBMS_OUTPUT.PUT_LINE('S-a modificat cel putin un rand.');
END IF;
```

Cursoare explicite

Acstea sunt definite de către programator și au astfel câteva caracteristici diferite față de cele prezentate mai sus. Ele trebuie definite cu nume obligatoriu și tipul returnat optional, iar deschiderea și închiderea lor este realizată tot de programator, spre deosebire de cele implicate. Cursoarele explicite se declară când o interogare SELECT returnează mai multe rânduri și dorim control asupra citirii lor.

```
DECLARE
    CURSOR c_nume IS
        continut_cursor;
BEGIN
    OPEN c_nume;
    LOOP
        FETCH c_nume INTO v_output;
        EXIT WHEN c_nume%NOTFOUND;
    END LOOP;
    CLOSE c_nume;
END;
```

Cursoare parametrizate

Un cursor poate primi parametri, asemenea unei funcții, pentru a filtra sau particulariza interogarea la momentul apelului. Această particularizare se realizează doar la nivelul parametrilor, nefiind posibilă o modificare structurală mai avansată decât alterarea unor valori. Astfel, se pot reutiliza aceleași definiții de cursor pentru criterii diferite:

```
DECLARE
    CURSOR c_nume (p_param1 VARCHAR2, p_param2 VARCHAR2) IS
        continut_cursor;
BEGIN
    OPEN c_nume('10B', 'Matematica');
    ...
END;
```

Cursoare FOR UPDATE

Aceste cursoare sunt folosite atunci când dorim să actualizăm sau să stergem rândurile citite. Prin adăugarea clauzei FOR UPDATE, cursorul blochează rândurile returnate pentru alte tranzacții, asigurând consistența datelor:

```
DECLARE
  CURSOR c_sal IS
    SELECT id_angajat, salariu FROM angajati
      FOR UPDATE OF salariu;
BEGIN
  FOR rec IN c_sal LOOP
    UPDATE angajati
      SET salariu = rec.salariu * 1.05
        WHERE CURRENT OF c_sal;
  END LOOP;
END;
```

Instrucțiunea WHERE CURRENT OF indică rândul curent al cursorului.

Cursoare dinamice (REF CURSOR)

Cursoarele dinamice, numite și REF CURSOR, oferă flexibilitate maximă, deoarece pot fi asociate la runtime cu interogări diferite. Aici se pot realiza schimbările la nivel structural care erau interzise în cadrul cursoarelor parametrizate, nefiind necesară vreo asemănare între cereri SQL asociate aceluiași REF CURSOR. Ele pot fi transmise ca parametri între proceduri sau funcții și pot returna rezultate variabile ca structură.

```
DECLARE
  TYPE ref_cur IS REF CURSOR;
  c_dyn ref_cur;
  v_nume elevi.nume%TYPE;
BEGIN
  OPEN c_dyn FOR 'SELECT nume FROM elevi WHERE varsta > 16';
```

```
LOOP
    FETCH c_dyn INTO v_nume;
    EXIT WHEN c_dyn%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(v_nume);
END LOOP;
CLOSE c_dyn;
END;
```

Atributele unui cursor

PL/SQL pune la dispoziție o serie de atribute predefinite pentru orice cursor, care oferă informații despre starea și conținutul acestuia. Atributele pot fi utilizate atât în cazul cursoarelor explicite cât și în cazul celor implicate (utilizând sintaxa SQL%atribut), însă doar în cadrul unui bloc procedural. Printre aceste atribute se numără:

- %FOUND – returnează TRUE dacă ultima operație FETCH a extras un rând valid; FALSE în caz contrar.
- %NOTFOUND – complementul atributului anterior; TRUE dacă nu mai există rânduri disponibile.
- %ROWCOUNT – indică numărul de rânduri extrase până la acel moment din cursor.
- %ISOPEN – returnează TRUE dacă cursorul este deschis, respectiv FALSE dacă a fost închis sau nu a fost deschis niciodată.

Aceste atribute se folosesc frecvent în bucle pentru a controla procesarea datelor extrase din cursor.

Exerciții

- a) Să se rețină toate informațiile despre angajații care lucrează într-un oraș cu numele dat de la tastatură. Să se afișeze pe ecran doar cele pentru primii 10 angajați în ordine alfabetică.
- b) Să se determine cel mai bine plătit angajat născut după anul 2000 și departamentul al cărui manager are cel mai mic salariu. Să se afișeze dacă angajatul determinat aparține departamentului determinat.
- c) Pentru fiecare angajat să se afișeze diferența dintre salariul lui actual și salariul maxim al jobului sau (găsit în câmpul max salary din tabelul JOBS)
- d) Să se afișeze numărul de angajați pentru departamentele localizate în UK și suma salariilor din departamentele localizate în US
- e) Să se mărească salariul cu 15% pentru fiecare angajat care lucrează în acele departamente care au manageri angrenați în proiecte.