

Baze de Date II

(Proiectarea Sistemelor cu Baze de Date)

Cornelia TUDORIE



Proiectarea Sistemelor cu Baze de Date - Cuprins

- A. Modelul Relațional.**
- B. Limbajul SQL. Lucrul cu tabele.**
- C. Sisteme cu baze de date. Performanța.**
- D. Obiecte ale sistemului de baze de date. Performanța în utilizare**
- E. Programe pentru baze de date. Performanța în programare.**
- F. Protecția datelor.**
- G. Sisteme Informatică. Proiectarea sistemelor de baze de date.**

E. Programe pentru baze de date. Performanța în programare.

- 1. Fișiere de comenzi**
- 2. PL/SQL. Tipuri de programe**
- 3. Blocuri anonime**
- 4. Subprograme stocate. Pachete**
- 5. Declanșatori**
- 6. Tranzactii**

E. 3. Blocuri anonime

- Bazele limbajului PL/SQL
- Accesul la baza de date
- Structuri de control
- Tipuri de date compuse
- **Cursori expliciți**
- Excepții

Cursori

Cursor - un nume asignat unei zone rezervate pentru executia unei cerere SQL.

doua tipuri: *cursor* - *implicit*
- *explicit*

Cursori

Cursor implicit - creat automat si utilizat de cate ori o cerere SQL apare in sectiunea executabila (cerere LMD si SELECT INTO, cand este returnata o singura linie!).

Cursor explicit - creat explicit si utilizat de cate ori o cerere SQL SELECT INTO returneaza mai multe linii.

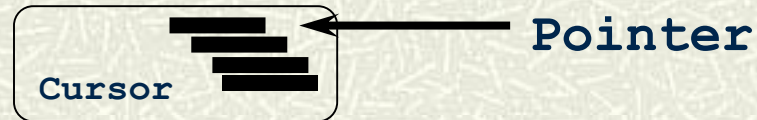
Cursori expliciti

Etape in utilizare:

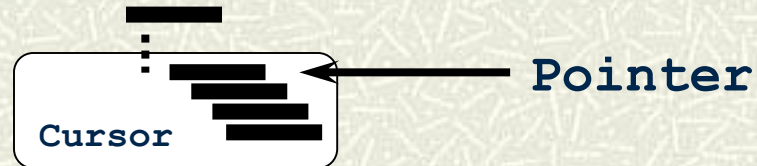
- *declarare (in sectiunea declarativa)*
- *deschidere (in sectiunea executabila)*
- *prelucrarea liniilor (in sectiunea executabila)*
- *inchidere (in sectiunea executabila)*

Cursori expliciti

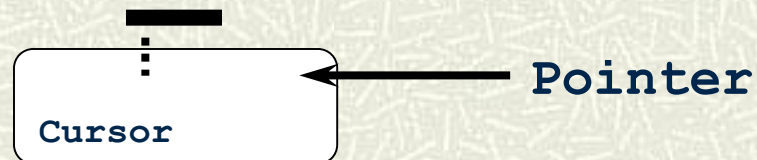
Deschidere



Incarcarea liniei
curente



Continua pana la
sfarsit



Inchidere



Cursori expliciti

Cursor

linia
curenta



10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Rezultat
(*result set*)

Cursori expliciti

Declarare:

```
CURSOR nume_cursor IS cerere ;
```

!! cerere FARA clauza INTO !!

Exemplu: **DECLARE**
 CURSOR c1 IS
 SELECT **ename, job**
 FROM emp WHERE deptno=10;

Cursori expliciti

Deschidere:

- *alocare de memorie*
- *analiza semantica si sintactica (parsing)*
- *executia cererii*
- *pozitionare pe prima linie*

Cursori expliciti

Deschidere:

```
OPEN nume_cursor ;
```

Exemplu:

```
OPEN c1;
```

Cursori expliciti

Inchidere:

- *eliberarea memoriei*

CLOSE *nume_cursor*

Exemplu:

CLOSE c1;

Cursori expliciti

Prelucrarea liniilor :

.... OPEN

.....

/ Prelucrarea liniilor :*

- structura repetitiva care include o instructiune FETCH*
- se pot folosi attributele cursorului*

**/*

.....

CLOSE

Cursori expliciti

Incarcarea liniei curente:

FETCH *nume_cursor* INTO *lista_variable* / *nume_record*

Cursori expliciti

Atributele unui cursor:

%ISOPEN

Boolean; TRUE daca cursorul este deschis

%FOUND

Boolean; TRUE daca linia a fost incarcata cu succes; FALSE daca nu este returnata nicio linie; NULL inaintea primei incarcari

%NOTFOUND

Boolean; TRUE daca nu este returnata nicio linie; FALSE daca linia a fost incarcata cu succes; NULL inaintea primei incarcari

%ROWCOUNT

Numarul de linii returnate de cerere (incarcate)

Cursori expliciti

Prelucrarea liniilor.

```
.... OPEN cursor ;  
...  
LOOP  
    FETCH cursor INTO variabile ;  
    EXIT WHEN ...;  
    ...  
        Prelucrari  
    ...  
END LOOP;  
...  
CLOSE ; .....
```


Cursori expliciti

Prelucrarea liniilor. Exemple

```
DECLARE
    v_empno                emp.empno%TYPE;
    v_ename                emp.ename%TYPE;
    i                      NUMBER := 1;
    CURSOR c1 IS
        SELECT empno , ename
        FROM emp;
BEGIN
    OPEN c1;
    FOR i IN 1..10 LOOP
        FETCH c1 INTO v_empno, v_ename;
        Dbms_output.put_line(to_char(i) || '. ' || v_ename);
    END LOOP;
    CLOSE c1;
END ;
```

Proiectarea Sistemelor cu Baze de Date -
Cornelia TUDORIE

Cursori expliciti

Prelucrarea liniilor. Exemple

```
DECLARE
    v_nrdep          dept.deptno%TYPE;
    v_ndep           dept.dname%TYPE;
    v_loc            dept.loc%TYPE;
    CURSOR c2 IS      SELECT * FROM dept;

BEGIN
    OPEN c2;
    FETCH c2 INTO v_nrdep, v_ndep, v_loc;
    WHILE (c2%FOUND) LOOP
        Dbms_output.put_line(v_ndep||' este situat la ' ||v_loc);
        FETCH c2 INTO v_nrdep, v_ndep, v_loc;
    END LOOP;
    CLOSE c2;

END ;
```

Proiectarea Sistemelor cu Baze de Date -
Cornelia TUDORIE

Cursori expliciti

Prelucrarea liniilor. Exemple

```
DECLARE
    v_nrdep          dept.deptno%TYPE;
    v_ndep           dept.dname%TYPE;
    v_loc            dept.loc%TYPE;
    CURSOR c3 IS      SELECT * FROM dept;

BEGIN
    OPEN c3;
    LOOP
        FETCH c3 INTO v_nrdep, v_ndep, v_loc;
        EXIT WHEN c3%NOTFOUND;
        Dbms_output.put_line(v_ndep||' este situat la ' ||v_loc);
    END LOOP;
    CLOSE c3;
END ;
```

Cursori expliciti

Prelucrarea liniilor. Exemple

```
DECLARE
    v_dep dept%ROWTYPE;
    CURSOR c4 IS SELECT * FROM dept;
BEGIN
    OPEN c4;
    FETCH c4 INTO v_dep;
    WHILE (c4%FOUND) LOOP
        Dbms_output.put_line(v_dep.dname ||
                               ' este situat la ' || v_dep.loc);
        FETCH c4 INTO v_dep;
    END LOOP;
    CLOSE c4;
END ;
```

Cursori expliciti

*Deschidere + Prelucrarea liniilor + Inchidere
(gestiune semi-automata)*

O bucla FOR :

- *inclocuieste OPEN, FETCH, CLOSE*
- *variabila de tip %ROWTYPE este automat definita*

Cursori expliciti

*Deschidere + Prelucrarea liniilor + Inchidere
(gestiune semi-automata)*

Exemplu:

```
DECLARE
v_dep dept%ROWTYPE;
CURSOR c5 IS SELECT * FROM dept;
BEGIN
OPEN c5;
FOR v_dep IN c5 LOOP
    Dbms_output.put_line(v_dep.dname || ' este situat la '
                        || v_dep.loc);
END LOOP;
CLOSE c5;
END ;
```


Cursori expliciti

Declarare + Deschidere + Prelucrarea liniilor + Inchidere
(declarare si gestiune semi-automata)

O bucla FOR :

- ***declararea se face prin instructiunea FOR***
- ***inclocuieste OPEN, FETCH, CLOSE***
- ***variabila de tip %ROWTYPE este automat definita***

!! Cursorul nu este reutilizabil !!

Cursori expliciti

*Declarare + Deschidere + Prelucrarea liniilor + Inchidere
(declarare si gestiune semi-automata)*

Exemplu:

```
DECLARE
CURSOR c5 IS SELECT * FROM dept;
BEGIN
  FOR v_dep IN (SELECT * FROM dept) LOOP
    Dbms_output.put_line(v_dep.dname || ' este situat la '
                        || v_dep.loc);
  END LOOP;
END ;
```

Cursori expliciti

Cursor parametrizat:

- *reutilizarea aceluiași cursor, cu diferiți parametri.*
- *transmiterea parametrilor se face la deschidere (OPEN) sau în buclă FOR*

!! Utilizarile se succed!!

Cursori expliciti

Cursor parametrizat:

CURSOR *nume_cursor* (*param1 type* [, *param2 type...*]) **IS** *cerere* ;

OPEN *nume_cursor* (*valoare1* [, *valoare2...*]);

FOR *var* **IN** *nume_cursor* (*valoare1* [, *valoare2...*]);

Cursori expliciti

Cursor parametrizat. Exemplu:

```
DECLARE
    CURSOR c6 (fctie varchar2) IS
        SELECT * FROM emp where job=fctie;
BEGIN
    Dbms_output.put_line('Analistii:');
    FOR v_emp IN c6('ANALYST') LOOP
        Dbms_output.put_line(v_emp.ename);

    END LOOP;

    Dbms_output.put_line('Managerii:');
    FOR v_emp IN c6('MANAGER') LOOP
        Dbms_output.put_line(v_emp.ename);

    END LOOP;
END ;
```


Cursori expliciti

Cursor explicit: permite parcurgerea
irversibila a rezultatului unei cereri

Tablou: permite recuperarea continutului
unui cursor si utilizarea lui pentru
prelucrari mai complexe

Cursori expliciti

Tablouri. Exemplu:

```
DECLARE
    TYPE depart IS TABLE OF dept.dname%type
        INDEX BY BINARY_INTEGER;
    tabnume depart;
    CURSOR c7 IS SELECT * FROM dept;
    i number :=1;
BEGIN
    FOR v_dep IN c7 LOOP
        tabnume(i) := v_dep.dname;
        i:=i+1;
    END LOOP;
    Dbms_output.put_line('al treilea departement este ' ||
        tabnume(3) );
END ;
```

Cursori expliciti

Avantaje:

- *efort redus de programare*
- *liniile sunt regasite si ordonate automat (intr-o maniere optimizata)*
- *dupa o modificare a structurii bazei de date, programul ramane valabil (in afara, eventual, de declararea cursorului)*

E. 3. Blocuri anonime

- Bazele limbajului PL/SQL
- Accesul la baza de date
- Structuri de control
- Tipuri de date compuse
- Cursori expliți
- **Excepții**

Excepții.

***Excepție** - un nume assignat unui eveniment care se produce în cursul execuției unui bloc*

Excepții.

Momente :

- *declarare (în secțiunea declarativă)*
- *producere (sau generare) (în secțiunea executabilă)*
- *tratare (într-o secțiune Exception sau în mediu)*

Excepții.

Tipuri de excepții:

predefinite, ORACLE

(ne-declarate;
generate de Oracle)

non-predefinite, ORACLE

(declarate;
generate de Oracle)

definite de utilizator

(declarate;
generate explicit)

Excepții.

Două situații când excepția este declarată :

- *pentru erori Oracle non-predefinite*
- *pentru excepții definite de utilizator*

Două modalități de declanșare a excepției:

- *automat, pentru erori Oracle*
- *explicit, prin instrucțiuni RAISE, în interiorul unui bloc*

Două modalități de tratare a unei excepții :

- *recuperare și rezolvare în secțiunea EXCEPTION a aceluiași bloc*
- *transmiterea la nivelul (blocul) superior (sau la mediu)*

Excepții. Declaraire

Excepții predefinite

Oracle

Nume excepție	Nr. Eroare
ACCESS_INTO_NULL	ORA-06530
COLLECTION_IS_NULL	ORA-06531
CURSOR_ALREADY_OPEN	ORA-06511
DUP_VAL_ON_INDEX	ORA-00001
INVALID_CURSOR	ORA-01001
INVALID_NUMBER	ORA-01722
LOGIN_DENIED	ORA-01017
NO_DATA_FOUND	ORA-01403
NOT_LOGGED_ON	ORA-01012
PROGRAM_ERROR	ORA-06501
ROWTYPE_MISMATCH	ORA-06504
STORAGE_ERROR	ORA-06500
SUBSCRIPT_BEYOND_COUNT	ORA-06533
SUBSCRIPT_OUTSIDE_LIMIT	ORA-06532
TIMEOUT_ON_RESOURCE	ORA-00051
TOO_MANY_ROWS	ORA-01422
VALUE_ERROR	ORA-06502
ZERO_DIVIDE	ORA-01476

Excepții. Declarare

Exemplu:

*în momentul când nicio linie nu este returnată de o comandă SELECT,
o eroare **ORA-01403** se produce și
excepția **NO_DATA_FOUND** va fi declanșată de
PL/SQL*

Excepții. Declaraire

*Excepțiile non-predefinite
(erori Oracle)*

Declaraire:

DECLARE

exception **EXCEPTION;**

PRAGMA EXCEPTION_INIT(exception, exception-code);

Excepții. Declaraire

Excepții definite de utilizator

Declaraire:

DECLARE

exception **EXCEPTION;**

Excepții. Declanșare

Declanșarea unei excepții:

- *automat, pentru erori Oracle*
- *explicit, prin instrucțiunea **RAISE**,
pentru o excepție definită de utilizator:*
RAISE exception;

Excepții. Tratare

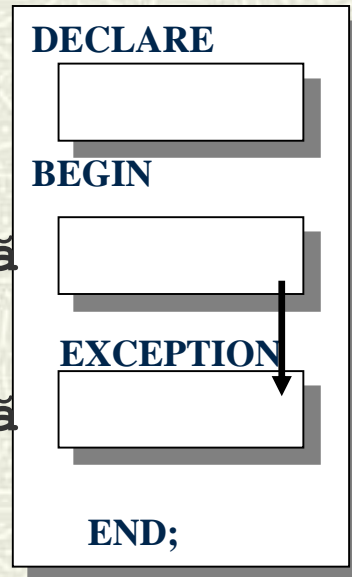
*Tratarea unei excepții:
recuperare și rezolvare, sau
transmiterea la blocul superior.*

Excepții. Tratare

Recuperarea sau transmiterea excepției

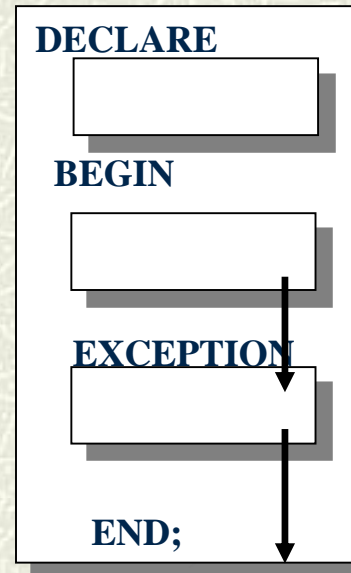
Excepția
este declanșată

Excepția
este recuperată



Excepția
este declanșată

Excepția
este transmisă



Excepții. Tratare

Recuperarea unei excepții:

EXCEPTION

WHEN *exception_1* [OR *exception_2* ...] THEN
***instructions_1*;**

[WHEN *exception_3* [OR *exception_4* ...] THEN
***instructions_2*; ...]**

[WHEN OTHERS THEN
***instructions_n*;]**

Excepții. Tratare

Recuperarea unei excepții:

- # **Toate excepțiile sunt recuperate când clauza `WHEN OTHERS` este prezentă. Niciuna nu este transmisă.**
- # **Excepțiile nerecuperate sunt transmise în absența clauzei `WHEN OTHERS`**

Excepții.

Exemplu excepții predefinite:

```
BEGIN
    .....
    SELECT . . . ;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        instructions1;
    WHEN TOO_MANY_ROWS THEN
        instructions2;
    WHEN OTHERS THEN
        instructions3;

END;
```


Excepții.

Exemplu excepții non-predefinite:

```
DECLARE
    e_err_dep  EXCEPTION;
    PRAGMA EXCEPTION_INIT (e_err_dep, -2292);

BEGIN
    .....

EXCEPTION
    WHEN e_err_dep THEN
        DBMS_OUTPUT.PUT_LINE('Departament inexistent');
    .....
END ;
```

Excepții.

Funcții utile pentru tratarea erorilor Oracle non-prededefinite:

- # **SQLCODE** (returnează codul numeric al erorii)
- # **SQLERRM** (returnează textul, sau mesajul erorii)

Excepții.

Funcții utile pentru tratarea erorilor Oracle non-prededefinite:

SQLCODE:

- 0 - nicio excepție**
- 1 - excepție definită de utilizator**
- 100 - excepție NO_DATA_FOUND**
- < 0 - erori Oracle**

Excepții.

Exemplu pentru excepții non-predefinite:

```
DECLARE
    e_err_cod  NUMBER;
    e_err_text VARCHAR2(30);
BEGIN
    .....
EXCEPTION
    WHEN OTHERS THEN
        e_err_cod:=SQLCODE;
        e_err_text:=SQLERRM;
        DBMS_OUTPUT.PUT_LINE('S-a produs eroarea nr ' ||
            e_err_cod || ' ; Ea spune: ' || e_err_text);
    .....
END ;
```


Excepții.

Exemplu pentru excepții definite de utilizator:

```
DECLARE
    e_err_sal_mare EXCEPTION;

BEGIN
    .....
    IF .... THEN
        RAISE e_err_sal_mare;
    .....
EXCEPTION
    WHEN e_err_sal_mare THEN
        DBMS_OUTPUT.PUT_LINE('Salariul este prea mare');
    .....
END ;
```