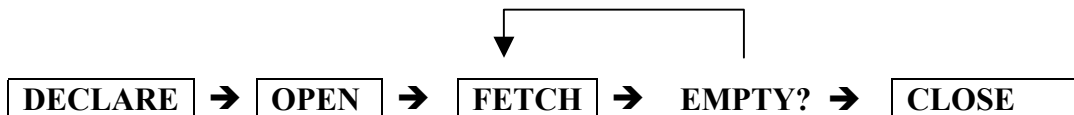


Lucrarea 10: Utilizare cursori

Orice comanda SQL executata pe serverul Oracle are asociat un cursor.
Asadar cursorii pot fi:

Impliciti: Sunt declarati de PL/SQL pentru orice SELECT chiar daca returneaza o singura linie.
Expliciti: Sunt declarati de catre programator si manipulati prin comenzi specifice in cadrul unui bloc PL/SQL
Controlul asupra unui cursor explicit se face de maniera urmatoare



* **CURSORS** * Programe pentru exemplificare utilizare cursor

*

```
Set pagesize 40
Set linesize 120
Set serveroutput ON
Set Verify off
```

* **Afisare cu LOOP EXIT WHEN c1%NotFound'**

DECLARE

```
-- Pentru controlul afisarii se foloseste functia CHR(10)-NL si CHR(13)-CR
VEmpno char(5);
VNumemp Scott.Emp.ename%Type;
VJob emp.job%type;
VDeptno emp.deptno%type;
```

CURSOR C1 IS Select EName,Empno,Job,Deptno From EMP;

BEGIN

OPEN C1; -- Deschidere cursor

```
DBMS_output.Put_line( 'Utilizare variabile declarate pentru campurile
cursor');
```

```
DBMS_output.Put_line(' Afisare cu LOOP ..... EXIT WHEN
c1%NotFound' ||CHR(10));
```

LOOP

FETCH C1 INTO VNumemp,VEmpno,VJob,VDeptno;

```
DBMS_output.Put_line(VNumemp||' '||VEmpno||' '||VJob||' '||VDeptno);
EXIT WHEN c1%NotFound;
```

END LOOP;

```
close c1; -- inchidere cursor
```

END;

/

```
COLUMN Ename HEADING 'Nume salariat' FORMAT A20
COLUMN Empno HEADING 'Marca' FORMAT 99999
COLUMN Deptno HEADING 'Nr.DPT' FORMAT 999
COLUMN Job HEADING 'Functia' FORMAT A10
```

```
SELECT * FROM Emp;
```

```
* Afisare cu For K IN 1..10 LOOP...EndLoop
```

```
DECLARE
```

```
VEmpno char(5);
```

```
VNume Scott.Emp.ename%Type;
```

```
VJob emp.job%type;
```

```
VDeptno emp.deptno%type;
```

```
CURSOR C1 IS Select EName,Empno,Job,Deptno From EMP;
```

```
BEGIN
```

```
OPEN C1; -- Deschidere cursor
```

```
DBMS_output.Put_line('Afisare cu For K IN 1..10 LOOP...EndLoop'||CHR(10));
```

```
DBMS_output.Put_line(' C1%rowcount: '||c1%rowcount );
```

```
For K IN 1..10 LOOP
```

```
FETCH C1 INTO VNume,VEmpno,VJob,VDeptno;
```

```
DBMS_output.Put_line(' C1%rowcount: '||c1%rowcount );
```

```
DBMS_output.Put_line(VNume||' '||VEmpno||' '||VJob||' '||VDeptno);
```

```
--EXIT WHEN c1%NotFound;
```

```
END LOOP;
```

```
close c1; -- inchidere cursor
```

```
END;
```

```
/
```

```
Utilizare WHILE C1%FOUND LOOP... End Loop
```

```
DECLARE
```

```
CURSOR C1 IS Select * From EMP;
```

```
R_c1 c1%RowType; -- Record avand coloanele cursorului C1
```

```
BEGIN
```

```
OPEN C1; -- Deschidere cursor
```

```
DBMS_output.Put_line('Utilizare Record R_C1 declarat de tipul cursorului  
c1%RowType'
```

```
||CHR(10)|| CHR(13)|| 'Afisare campuri cursor cu R_C1.camp||
```

```
CHR(10)|| CHR(13)|| ' Utilizare WHILE C1%FOUND LOOP... End Loop '||CHR(10));
```

```
--For K IN 1..10 LOOP
```

```
IF C1%FOUND THEN
```

```
DBMS_output.Put_line('C1%FOUND ');
```

```
ELSE
```

```
DBMS_output.Put_line('ATENTIE: C1%NotFOUND= true inainte de FETCH '
```

```
||CHR(10));
```

```
END IF;
```

```
FETCH C1 INTO R_c1; -- salvare inreg. curenta in R_c1
```

```
WHILE C1%FOUND LOOP
```

```
-- DBMS_output.Put_line(' C1%rowcount: '||c1%rowcount );
```

```
DBMS_output.Put_line(C1%rowcount||' '||R_c1.EName||' '||R_c1.Empno||
```

```
' '||R_c1.Job||' '||R_c1.Deptno);
```

```

FETCH C1 INTO R_c1; -- salvare inreg. curenta in R_c1
END LOOP;
close c1; -- inchidere cursor
END;
/

```

*** Utilizare deschidere implicita cursor**

```

DECLARE
CURSOR C1 IS Select * From EMP;
R_c1 c1%RowType; -- Record avand coloanele cursorului C1

```

```

BEGIN
-- OPEN C1; -- Deschidere cursor
DBMS_output.Put_line('Utilizare Record R_C1 declarat de tipul
cursorului c1%RowType' ||CHR(10)|| CHR(13)||
'Afisare campuri cursor cu R_C1.camp' || CHR(10)|| CHR(13)||
' Utilizare LOOP implicit pentru cursor' ||CHR(10)|| CHR(13)||
'FOR R_c1 IN c1 LOOP... End Loop' ||CHR(10));

--FETCH C1 INTO R_c1; -- salvare inreg. curenta in R_c1
-- WHILE C1%FOUND LOOP
FOR R_c1 IN C1 LOOP -- LOOP implicit pentru cursor
DBMS_output.Put_line(C1%rowcount||' ' ||R_c1.EName||' ' ||R_c1.Empno||' '
||R_c1.Job||' ' ||R_c1.Deptno);
-- FETCH C1 INTO R_c1; -- salvare inreg. curenta in R_c1
END LOOP;
--close c1; -- inchidere cursor
END;
/

```

**** CURSOR2**

```

* Actualizare tabela folosind CURSOR cu FOR UPDATE OF cimp
* si UPDATE tabela SET cimp=expr WHERE CURRENT OF cursor
SET SERVEROUTPUT ON
SET PAGESIZE 40
SET LINESIZE 120

```

```

DECLARE
CURSOR C1 IS Select * From pers FOR UPDATE OF sal;
R_c1 c1%RowType; -- Record avand coloanele cursorului C1
R_c2 c1%RowType; -- Record 2
BEGIN

```

```

DBMS_output.Put_line('Utilizare Record R_C1 declarat de tipul cursorului
c1%RowType' ||CHR(10)|| CHR(13)||
'Modificare campuri cursor declarat cu FOR UPDATE referite cu
R_C1.camp' ||CHR(10)|| CHR(13)||
' Utilizare FOR R_c1 IN c1 LOOP... End Loop ' ||CHR(10)||
' si UPDATE.... WHERE CURRENT OF c1' ||CHR(10));

```

```

-- Afisare valoare veche si actualizare
FOR R_c1 IN c1 LOOP -- LOOP implicit pentru cursor
-- R_c1 indica masca si var.record unde se salveaza inregistrarea
DBMS_output.Put_line(C1%rowcount||' ||R_c1.ENAME||' ||R_c1.Empno||
' ||R_c1.Job||' ||R_c1.Deptno||' Salar: ||R_c1.sal);
UPDATE pers set sal= sal*1 WHERE CURRENT OF c1;
R_C2:=R_C1; -- atribuire pentru record2
exit when C1%rowcount>7;
END LOOP;
DBMS_output.Put_line(chr(10));

-- Afisare inregistrari dupa actualizare
FOR R_c1 IN c1 LOOP
DBMS_output.Put_line(C1%rowcount||' ||R_c1.ENAME||' ||R_c1.Empno||
' ||R_c1.Job||' ||R_c1.Deptno||' Salar: ||R_c1.sal);
exit when C1%rowcount>5;
END LOOP; -- inchide cursorul

-- Afisare inregistrare salvata in R_c2
DBMS_output.Put_line(CHR(10)||'Record R_C1'||' ||R_c1.ENAME||'
||R_c1.Empno||
' ||R_c1.Job||' ||R_c1.Deptno||' Salar: ||R_c1.sal);
DBMS_output.Put_line('Record R_c2 ||R_c2.ENAME||' ||R_c2.Empno||
' ||R_c2.Job||' ||R_c2.Deptno||' Salar: ||R_c2.sal);
END;
/

* CURSOR 3
* Tabela de record-uri pentru salvare inreg. din EMP
* Utilizare deschidere implicita cursor
SET VERIFY OFF
Set pagesize 40
set linesize 120
Set serveroutput ON

DECLARE
K INTEGER; -- Indice in ciclu LOOP
KM INTEGER; -- Nr. de inregistrari citite
CURSOR C1 IS Select * From EMP;
R_c1 c1%RowType;-- Record avand coloanele cursorului C1
TYPE TR IS TABLE OF Emp%RowType index by binary_integer;
TR_c1 TR; -- Tabela de record-uri
BEGIN
OPEN C1; -- Deschidere cursor
DBMS_output.Put_line('* Tabela de record-uri pentru salvare inreg. din
EMP'||CHR(10)||
'Utilizare Record R_C1 declarat de tipul cursorului c1%RowType' ||CHR(10)||
'Afisare campuri cursor cu TR_C1(K).camp'|| CHR(10)||

```

```

' Utilizare FOR K IN 1..N LOOP' || CHR(10));

FOR K IN 1..&N LOOP    -- LOOP implicit pentru cursor
FETCH C1 INTO TR_c1(K);    -- salvare inreg. in TR_c1
-- FETCH C1 INTO R_c1;    -- varianta
-- TR_C1(K):=R_c1;
EXIT WHEN c1%NotFound;
KM:=k;    -- nr de inregistrari
END LOOP;

close c1;    -- inchidere cursor
DBMS_output.Put_line(' Salariati: '||KM||CHR(10));

-- Afisare tabelul de record-uri
FOR K IN 1..KM LOOP
DBMS_output.Put_line(K||' '||TR_c1(K).ENAME||' '||TR_c1(K).EMPNO||' '
||TR_c1(K).JOB||' '||TR_c1(K).DEPTNO||' '||TR_c1(K).SAL);
END LOOP;
END;
/

* Cursor cu parametrii
DECLARE
K INTEGER; -- Indice in ciclu LOOP
KM INTEGER;    -- Nr. de inregistrari citite
CURSOR C1 (vjob emp.job%Type) IS Select * From EMP
WHERE job=vjob;
R_c1  c1%RowType;-- Record avand coloanele cursorului C1
TYPE TR IS TABLE OF Emp%RowType index by binary_integer;
TR_c1 TR;    -- Tabela de record-uri
BEGIN
OPEN C1('&Job');    -- Deschidere cursor
DBMS_output.Put_line('* Tabela de record-uri pentru salvare inreg. din
EMP'||CHR(10)||
'Utilizare Record R_C1 declarat de tipul cursorului c1%RowType' ||CHR(10)||
'Afisare campuri cursor cu TR_C1(K).camp'|| CHR(10)||
' Utilizare FOR K IN 1..N LOOP' || CHR(10));

FOR K IN 1..100 LOOP    -- LOOP implicit pentru cursor
FETCH C1 INTO TR_c1(K);    -- salvare inreg. in TR_c1
-- FETCH C1 INTO R_c1;    -- varianta
-- TR_C1(K):=R_c1;
EXIT WHEN c1%NotFound;
KM:=k;    -- nr de inregistrari
END LOOP;

close c1;    -- inchidere cursor
DBMS_output.Put_line(' Salariati: '||KM||CHR(10));

-- Afisare tabelul de record-uri

```

FOR K IN 1..KM LOOP

DBMS_output.Put_line(K||' '||TR_c1(K).ENAME||' '||TR_c1(K).Empno||' '
||TR_c1(K).Job||' '||TR_c1(K).Deptno||' '||TR_c1(K).Sal);

END LOOP;

END;

/