Lucrarea 11.

Definire si utilizare subprograme in PL/SQL

Subprogramele create se compileaza si se depun ca obiecte in baza de date. Ele pot fi apelate de orice utilizator fiindca nu apartin programului ci sistemului BD. La apelul subprogramului i se transmit doar parametrii de pe statia client, incarcarea facandu-se din BD de pe server in memorie unde se lanseaza.
Un program incarcat in memoria poate fi refolosit de alti utilizatori direct din memorie cache din zona "Sharing pool" . Un subprogram se pastreaz intr-o singura copie in memorie indiferent de numarul de utilizatori care il apeleaza. Fiecarui utilizator i se aloca o zona pentru parametrii codul programului fiind reentrant.

Avantajele utilizarii subprogramelor sunt cunoscute:

- Programare modulara cu facilitati de a lucra in echipa pe task-uri
- Productivitatea programarii ridicata
- Intretinere si depanare usoara (functia neeexecutata se cauta intr-o procedura)
- Facillitati de dezvoltare a aplicatiei prin adaugare proceduri pentru functii noi
- Programul usor de urmarit apeland subprograme din programul principal
- Performata ridicata prin pastrarea subprogramelor in memorie
- Creste securitatea BD prin posibilitatea de a da drepturi de doar de executie unui utilizator pe un subprogram si nu drepturi de acces la tabelele BD.
- Subprogramul poate lucra cu drepturile utlizatorului a celui care l-a definit
- Subprogramului i se transmit doar parametrii spre server nu si codul ca la blocurile PL/SQL, care trebuie si compilat.
- Extinderea setului de functii si proceduri sistem

Subprogramele admise de PL/SQL sunt **PROCEDURE si FUNCTION**
Sintaxa comenzii pentru crearea une proceduri permite specificarea explicita a parametrilor de intrare IN si iesire OUT sau de intrare iesire IN OUT :

```
[CREATE [OR REPLACE]                IN                          CURENT_USER
PROCEDURE nume _procedura (par1 OUT tip [:=expr] [AUTHID  DEFINER    ]
                                    IN OUT
AS
        Var1    tip1;               -- declaratii variabile, recorduri si cursoare locale
        Var2    tip2;
        Cursor c1 AS SELECT.....;
BEGIN
        Comenzi executabile...
.......
EXCEPTION
        Comenzi tratare erori ...
        .......
END nume_procedura;
/
```

Procedurile fara CREATE sunt proceduri standalone si nu se memoreaza.
O comanda de definire procedura se comporta ca un bloc PL/SQL cu nume.

Sintaxa comenzii de creare a unei functii este asemanatoare cu cea procedurii dar ea returneaza o singura valoare si poate fi utilizata ca operator in expresii:

```
[CREATE [OR REPLACE]                                    CURENT_USER
FUNCTION nume _functie (par1 tip [:=expr] [AUTHID  DEFINER        ]
RETURN tip_functie
AS
        Var1    tip1;           -- declaratii variabile, recorduri si cursoare locale
        Var2    tip2;
        Cursor c1 AS SELECT.....;
BEGIN
        Comenzi executabile...
.......
RETURN expr;
EXCEPTION
        Comenzi tratare erori ...
        .......
END nume_procedura;
/
```

*** PROCED1 * Program pentru definire proceduri**

```
SET serveroutput on
```

*** AFIS_SAL * Procedura pentru afisare salar si impozit**

```
* Se da codul salariatului -Empno

Create or Replace procedure Afis_sal
 (cods IN integer, salar OUT  Number,impozit OUT Number)
 AS
mes Varchar2(20) := ' Salarul brut este: ';
Cursor c1 IS SELECT * from EMP;
R1 EMP%ROWTYPE;
Vsalar Number(7);
Vimpozit Number(5);
BEGIN
FOR R1 IN c1 LOOP
IF r1.empno=cods THEN
Vsalar:=r1.sal +NVL(r1.comm,0);              -- salar brut
Vimpozit:=Vsalar*0.25;
impozit:=Vimpozit;              -- impozit
salar:=Vsalar - Vimpozit;       -- salar net
DBMS_OUTPUT.PUT_LINE(mes|| Vsalar|| '  Impozitul:  '||Vimpozit ||
' Nume: '|| r1.ename);
End If;
END LOOP;
END afis_sal;
/
```

*** AFIS_nume * Procedura pentru afisare salar si impozit**

```
* Se da nume salariat -Ename
```

Create or Replace procedure Afis_nume
(nume IN Varchar2, salar OUT Number,impozit OUT Number)
AS

```
mes Varchar2(20) := ' Salarul brut este: ';
Cursor c1 IS SELECT * from EMP;
R1 EMP%ROWTYPE;
Vsalar Number(7);
Vimpozit Number(5);
BEGIN
FOR R1 IN c1 LOOP
IF r1.ename=nume THEN
Vsalar:=r1.sal +NVL(r1.comm,0);                -- salar brut
Vimpozit:=Vsalar*0.25;
impozit:=Vimpozit;              -- impozit
salar:=Vsalar - Vimpozit;       -- salar net
DBMS_OUTPUT.PUT_LINE(mes|| Vsalar|| '  Impozitul:  '||Vimpozit ||
' Nume: '|| r1.ename);
End If;
END LOOP;
END afis_nume;
/
```

*** Sal_med * Functie pentru afisare salar mediu pe departament**

Create or Replace FUNCTION Sal_med (NrDept Number) RETURN NUMBER
AS

```
mes Varchar2(20) := ' Salarul mediu este: ';
Cursor c1 IS SELECT * from EMP Where deptno=NrDept;
R1 EMP%ROWTYPE;
Vsalar Number(7);
Vimpozit Number(5);
Vsuma Number(7):=0;          -- Suma salarii
N        Number(3):=1;          -- Numar salariati
Vmed  Number(8);     -- Medie salarii
BEGIN
Vsuma:=0;
N:=1;
FOR R1 IN c1 LOOP
Vsuma:=Vsuma+r1.sal;
N:=N+1;
DBMS_OUTPUT.PUT_LINE(' Suma:  '||Vsuma|| ' pentru departamentul:  '||NrDept);
END LOOP;
Vmed:=Vsuma/N;                 -- Media salarii
DBMS_OUTPUT.PUT_LINE(mes||Vmed|| ' pentru departamentul:  '||NrDept);
RETURN Vmed;
END Sal_med;
/
```

*** CAL_P * Program pentru chemare proceduri**

```
set serveroutput on
set linesize 120
set pagesize 40
set verify off
Variable Nume1 varchar2(10)
DECLARE
Vsalar NUMBER(7);
Vimpozit Number(5);
VNume varchar2(10):='&nume_salariat';      --Nu trebuie apostroafe la introducere
R         Number(8);     -- rezultat expresie cu functii
BEGIN
:Nume1:='&Nume1';
afis_sal(&Cod_salariat,Vsalar,Vimpozit);
dbms_output.put_line('  Salar:  '||Vsalar|| '  Impozit:  '||Vimpozit);
afis_sal(7900,Vsalar,Vimpozit);
dbms_output.put_line('  Salar:  '||Vsalar|| '  Impozit:  '||Vimpozit);
Afis_nume(VNume, Vsalar,Vimpozit);
dbms_output.put_line('  Salar:  '||Vsalar|| '  Impozit:  '||Vimpozit||
          '  Nume salariat:  '||Vnume) ;
Afis_nume(:Nume1, Vsalar,Vimpozit);
dbms_output.put_line('  Salar:  '||Vsalar|| '  Impozit:  '||Vimpozit||
          '  Nume salariat:  '||:nume1) ;
-- Apelare functie Sal_med
r:=SAL_MED(20)+SAL_MED(30);
dbms_output.put_line('  Salar mediu:  '||r);
end;
/
```