UNIUNEA EUROPEANĂ

GUVERNUL ROMÂNIEI

Instrumente Structurale
2007-2013

Platformă de e-learning și curriculă e-content pentru învățământul superior tehnic

Proiect nr. 154/323 cod SMIS – 4428 cofinanțat de prin Fondul European de Dezvoltare Regională “Investiții pentru viitorul dumneavoastră”.

Programul Operaţional Sectorial Creşterea Competitivităţii Economice - POS CCE

UNIUNEA EUROPEANĂ

GUVERNUL ROMÂNIEI

Instrumente Structurale
2007-2013

Platformă de e-learning și curriculă e-content pentru învățământul superior tehnic

Baze de date

27. Funcții definite de utilizator

Introducere

O funcție este un subprogram care acceptă parametri, poate fi apelată dintr-un program apelant şi returnează o valoare. În general, funcțiile se folosesc pentru a procesa un set de date şi returna un rezultat. Funcțiile şi procedurile sunt structuri asemănătoare. O funcție poate returna o singură valoare, în timp ce o procedură poate returna una sau mai multe valori prin intermediul parametrilor *OUT*. Funcțiile declarate în cadrul unui bloc sunt funcții stocate în memorie împreună cu blocul PL/SQL în care sunt declarate şi au o funcționalitate asemănătoare procedurilor. Durata lor de viață este cât a blocului şi nu pot fi apelate decât din blocul respectiv. După închiderea sesiunii, funcțiile nestocate pe server se pierd, deoarece nu au fost salvate în dicționarul bazei de date. Pentru a fi recreată o astfel de funcție, se compilează din nou, împreună cu blocul în care a fost declarată. Funcțiile pot fi folosite în blocul apelant folosind operatorul de atribuire sau în cadrul unor expresii şi condiții. Funcțiile stocate pe server sunt funcții care sunt create în dicționarul bazei de date şi pot fi accesate prin comenzi SQL ca orice obiect, dacă utilizatorul are suficiente privilegii. Parametrii unei funcții respectă aceleaşi reguli ca în cazul procedurilor.

Funcții declarate în cadrul unui bloc PL/SQL

```
DECLARE                                          -- secțiunea de declarare a funcției
FUNCTION function_name                           -- declararea numelui funcției[
parameter_name [IN |OUT |IN OUT] parameter_type, … ] RETURN data_type IS
        [function _declaration_section]
BEGIN                                            -- blocul funcției
        function _executable_section
[EXCEPTION
        function _exception_section]
END [function _name];                            -- sfârşitul declarării funcției
BEGIN                                            --  blocul apelant
DECLARE
block_variables;
BEGIN
            block_ executable_section
            var:=function_name[ parameters];  -- apelul funcției
[EXCEPTION
  block exceptions]
        END;
END;                                                 -- sfârşitul blocului apelant
```

Cuvinte cheie

- ***function_name*** – este numele funcției;
- ***parameter_name*** – reprezintă numele unui parametru formal din lista de parametri;
- ***parameter_type*** – reprezintă tipul unui parametru formal;
- ***function_declaration_section*** – reprezintă secțiunea de declarare de variabile pentru funcție;
- ***function_executable_section*** – reprezintă secțiunea executabilă a funcției;
- ***function_exception_section*** – reprezintă secțiunea de tratare a excepțiilor din cadrul funcției;
- ***block_variables*** - reprezintă secțiunea de declarare de variabile pentru blocul apelant;
- ***var*** – este numele variabilei în care funcția returnează o valoare;
- ***block exceptions*** – reprezintă secțiunea de tratare a excepțiilor în cadrul blocului apelant;
- ***IN |OUT |IN OUT*** – specifică dacă parametrul poate fi referit sau modificat în interiorul sau exteriorul funcției.

Exemplu:

```
DECLARE  FUNCTION  salariati (dep NUMBER) RETURN NUMBER IS
nr_ang NUMBER;
BEGIN
  SELECT count(distinct  id_ang) INTO nr_ang FROM angajati  WHERE id_dep = dep ;
RETURN nr_ang;
END;
BEGIN
DECLARE
depart VARCHAR2(15); nr_dep number(2); nr_sal number(5);
BEGIN
        nr_dep:=&nr_depart;
        SELECT den_dep INTO depart FROM departamente     WHERE id_dep=nr_dep;
        nr_sal:=salariati(nr_dep);
        dbms_output.put_line ('Departamentul'||' '||rpad(depart,15)  ||' '||'are '||
        lpad(nr_sal, 5)||' salariati.' );
EXCEPTION
        WHEN NO_DATA_FOUND THEN  dbms_output.put_line  ('Departament inexistent');
END;
END;
```

Funcții PL/SQL stocate pe server

Sintaxa de creare este următoarea:

```
CREATE [OR REPLACE]  FUNCTION  function_name
        [ parameter_name [IN |OUT |IN OUT]

        parameter_type, … ] RETURN variable_type

  [AUTHID {DEFINER | CURRENT_USER}]

  [PRAGMA AUTONOMOUS_TRANSACTION]
  IS /AS
      [declaration_section]
  BEGIN
      executable_section
  [EXCEPTION
      exception_section]
  END [function_name];
```

Cuvinte cheie

- ***function_name*** – este numele funcției;
- ***parameter_name*** – reprezintă numele unui parametru formal din lista de parametri;
- ***parameter_type*** – reprezintă tipul unui parametru formal;
- ***declaration_section*** – reprezintă secțiunea de declarare de variabile ;
- ***executable_section*** – reprezintă secțiunea executabilă a funcției ;
- ***exception_section*** – reprezintă secțiunea de tratare a excepțiilor ;
- ***[AUTHID {DEFINER | CURRENT_USER}]*** - specifică dacă o funcție stocată se execută cu drepturile celui care a creat-o (valoare implicită) sau ale utilizatorului curent;
- ***[PRAGMA AUTONOMOUS_TRANSACTION]*** - specifică că execuția funcției suspendă tranzacția curentă, care se reia după terminarea execuției funcției, adică într-o tranzacție imbricăm altă tranzacție, cu propriile sale *COMMIT* sau *ROLLBACK;*
- ***IN |OUT |IN OUT*** – specifică dacă parametrul poate fi referit sau modificat în interiorul sau exteriorul funcției.

Exemplu:

```
CREATE OR REPLACE FUNCTION puncte(ecuson in number) RETURN number AS
        data_ang angajati.data_ang%TYPE;
        dep number(2); sal number; com number; sal_max number; pcte number := 0;
BEGIN
SELECT data_ang, id_dep, salariu, nvl(comision, 0) INTO data_ang, dep, sal, com
FROM angajati WHERE id_ang = ecuson;
        IF months_between(sysdate, data_ang) > 25*12 THEN
                pcte := pcte + 30;
        END IF;
SELECT max(salariu) INTO sal_max FROM angajati WHERE id_dep = dep;
        IF sal_max = sal  THEN    pcte := pcte + 20;   END IF;
        IF com > 0    THEN    pcte := pcte + 10;    END IF;
RETURN pcte;
END  puncte;
```

Programul apelant

```
DECLARE
        pcte number;
        cursor c_ang IS
                SELECT id_ang, nume, id_dep FROM angajati;
 BEGIN
        DELETE FROM temp_puncte;
        FOR i IN c_ang
           LOOP
                pcte := puncte(i.id_ang);
                INSERT INTO temp_puncte  VALUES (i.id_ang, pcte, i.id_dep);
           END LOOP;
END;
```

Apelul funcțiilor din comenzi SQL

- Odată create şi stocate în baza de date, funcțiile pot fi apelate în comenzile SQL, dar cu anumite restricții.
- Dacă funcția a fost deja creată în dicționarul bazei de date, nu se mai poate crea cu acelaşi nume. În acest caz, trebuie folosită comanda *CREATE OR REPLACE FUNCTION* şi sistemul de gestiune va înlocui în dicționar vechea funcție cu noua funcție, dar cu acelaşi nume. O altă posibilitate este ştergerea din dicționar, cu comanda:
- ***SQL> DROP FUNCTION function_name;***
- apoi se crează din nou, cu comanda *CREATE FUNCTION*.
- Dacă funcția respectivă este definită în cadrul unui pachet, atunci ştergerea ei implică redefinirea pachetului şi eliminarea specificațiilor legate de aceasta. Daca se şterge tot pachetul se vor şterge automat toate procedurile şi funcțiile definite în cadrul lui şi nu mai este nevoie de comenzi suplimentare.

Apelul funcțiilor din comenzi SQL

- Funcția **PUNCTE**, creată anterior, care returnează numărul de puncte obținute de un angajat, în vederea întocmirii unui clasament pe departament, poate fi folosită astfel:
- **SQL> SELECT a.den_dep, b.nume, PUNCTE(b.id_ang) punctaj**

  FROM departamente a, angajati b

  WHERE a.id_dep=b.id_dep AND a.id_dep=10;

- **SQL>SELECT a.den_dep, b.nume, PUNCTE(b.id_ang) punctaj**

  FROM departamente a, angajati b

  WHERE a.id_dep=b.id_dep AND PUNCTE(b.id_ang)>=50;

Informații din dicționarul bazei de date

Codul sursă al procedurilor şi funcțiilor stocate se găseşte în dicționarul bazei de date, în tabela *USER_SOURCE*, care are următoarea structură:

- **SQL> desc user_source**

Name	Null?	Type
NAME		VARCHAR2(30)
TYPE		VARCHAR2(12)
LINE		NUMBER
TEXT		VARCHAR2(4000)

unde:

name – este numele procedurii sau funcției;

type – reprezintă tipul obiectului;

line - este numărul liniei de cod;

text – este codul sursă din linia respectivă.

- **SQL> SELECT text FROM user_source WHERE name = 'PUNCTE‘ ORDER BY line;**

Considerente asupra procedurilor şi funcțiilor

- În general, o procedură este folosită pentru secvențe de cod mai complexe, în timp ce o funcție este folosită în cadrul procedurii pentru calcule repetate şi mai simple.
- Din punct de vedere tehnic, sunt câteva mici diferențe între proceduri şi funcții:
- o funcție returnează totdeauna o valoare în programul apelant;
- o procedură nu returnează direct o valoare, dar poate întoarce mai multe valori în programul apelant, prin parametrii de tip *OUT*;
- funcțiile stocate pot fi apelate direct din comenzi SQL;
- procedurile stocate nu pot fi apelate direct din comenzi SQL ;
- funcțiile pot fi folosite în operații de atribuire, în expresii şi condiții pe clauza WHERE;
- procedurile nu pot fi folosite în operații de atribuire, expresii şi condiții pe clauza WHERE.

Considerente asupra procedurilor şi funcțiilor-continuare

- Dacă într-o procedură sau funcție apare o excepție, tratată de sistem sau definită de utilizator, controlul este preluat de secțiunea *EXCEPTION* sau de blocul superior, iar parametrilor de tip *OUT* sau *IN OUT* nu li se transmite nicio valoare. În acest caz, parametrii actuali din procedura sau funcția apelantă vor pastra aceeaşi valoare pe care au avut-o înainte de a se face apelul.
- Execuția comenzilor *CREATE OR REPLACE* sau *DROP* pentru proceduri şi funcții şi a comenzilor *ALTER TABLE* sau *ALTER VIEW* pentru tabele şi view-uri, pot modifica starea altor obiecte din baza de date cu care sunt relaționate.
- Sistemul de gestiune Oracle permite acordarea privilegiului de execuție asupra procedurilor si funcțiilor şi altor utilizatori, acțiune permisă proprietarului lor(userul care le-a creat) sau administratorului care are privilegiu de *DBA*. Comanda SQL este:
- ***SQL> GRANT EXECUTE ON procedure_name/function_name***

  TO user_name;
- Stergerea privilegiului se face cu comanda *REVOKE.*