



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculum 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 Operațional Sectorial Creșterea Competitivității Economice - POS CCE



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



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

Sisteme CAD/CASE

25. Funcții și proceduri în Forms Builder

Introducere

- Funcțiile sunt subprograme care manipulează și prelucrează date și returnează un rezultat. Procedurile sunt tot subprograme, folosite pentru prelucrarea datelor, dar care nu trebuie să returneze direct un rezultat, însă rezultatele pot fi folosite în programul apelant. Funcțiile și procedurile PL/SQL acceptate în Forms Builder sunt de doua feluri:
 - Predefinite (sau de sistem);
 - Definite de utilizator.

Utilizarea funcțiilor și procedurilor oferă următoarele avantaje:

- Ușurează și simplifică proiectarea aplicațiilor;
- Se controlează mai ușor funcționalitățile cerute în specificații;
- Cresc performanțele de funcționare a aplicațiilor;
- Pot fi executate la apariția anumitor evenimente monitorizate de triggeri;
- Optimizează alocarea spațiului de memorie, deoarece beneficiază de caracteristicile mediului Oracle;
- Îmbunătățesc integritatea și coerența datelor;
- Se reduce traficul în rețea, deoarece sunt stocate în memorie.

Funcții și proceduri de sistem

- Sunt funcții /proceduri stocate în biblioteca de sistem și pot fi apelate în timpul execuției unui formular. Câteva exemple de funcții și proceduri:
- **SET_WINDOW_PROPERTY** - setează proprietățile unei ferestre.
- **SET_ITEM_PROPERTY** - modifică toate instanțele unui element dintr-un bloc, prin modificarea unei proprietăți a elementului respectiv.
- **MESSAGE** - afișează textul specificat în linia de mesaje.
- **NEXT_RECORD** - permite navigarea la primul element din următoarea înregistrare (permite navigarea de la o înregistrare la alta).
- **PREVIOUS_RECORD** - permite navigarea la primul element din înregistrarea anterioară
- **FIRST_RECORD** - permite navigarea la prima înregistrare din bloc.
- **LAST_RECORD** - permite navigarea la ultima înregistrare din blocul curent.
- **CREATE_RECORD** - creează o nouă înregistrare în blocul curent, după înregistrarea curentă.
- **INSERT_RECORD** - se utilizează numai într-un trigger ON-INSERT și inserează o linie în baza de date, după salvarea înregistrării(*commit*).
- **EXIT_FORM** – încheie execuția unui formular și confirmă acțiunea de commit.

Funcții și proceduri utilizator

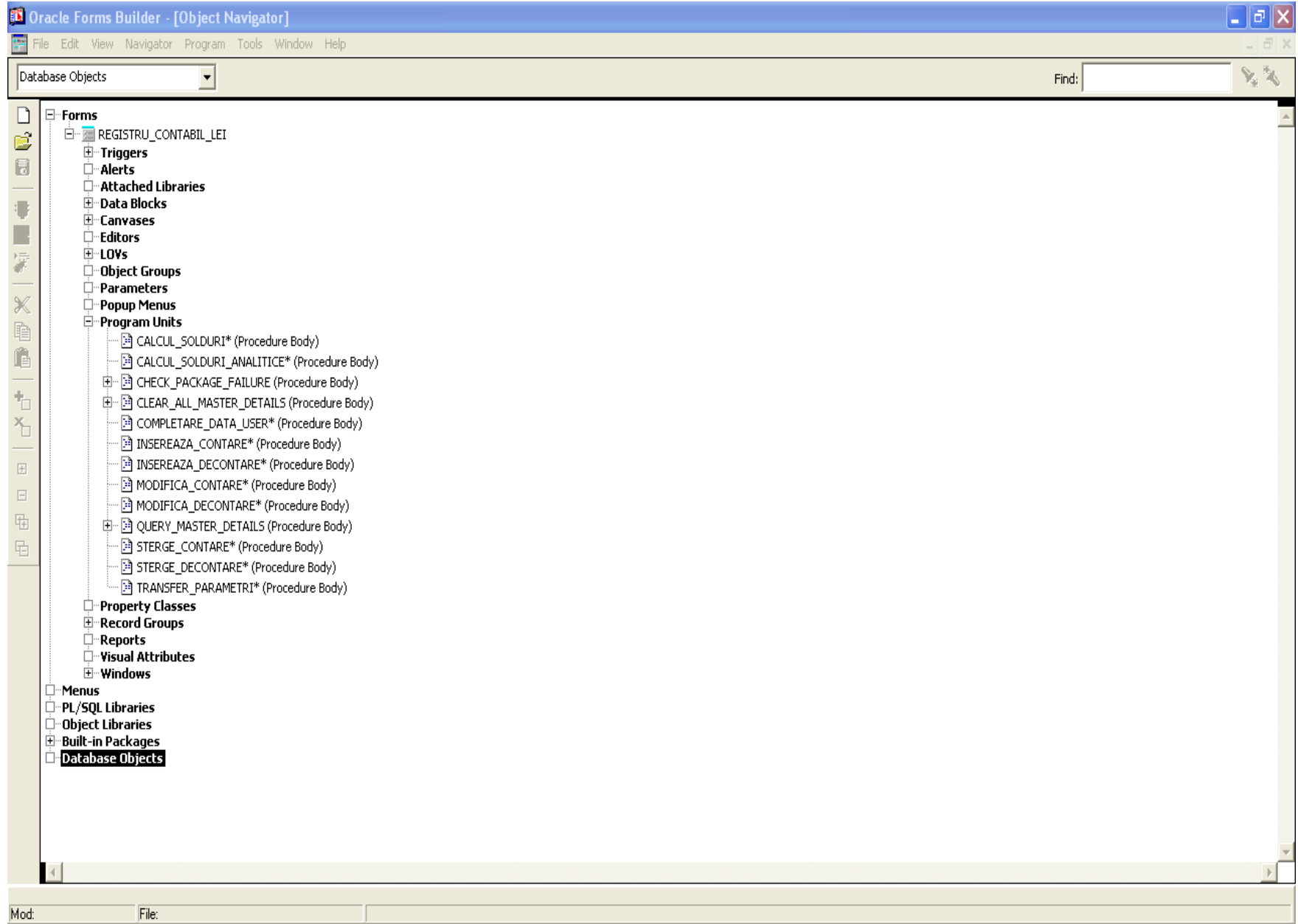
- Sintaxa unei funcții PL/SQL în Forms Builder este următoarea:
- **FUNCTION** *function_name* -- declararea numelui funcției
 [*parameter_name* [IN | OUT | IN OUT] *parameter_type*, ...] **RETURN**
 data_type **IS**
 [*function_declaration_section*]
BEGIN -- blocul funcției
 function_executable_section
 [**EXCEPTION**
 function_exception_section]
END [*function_name*]; -- sfârșitul declarării funcției
- Sintaxa unei proceduri PL/SQL în Forms Builder este următoarea:
- **PROCEDURE** *procedure_name* -- declararea procedurii
 [*parameter_name* [IN | OUT | IN OUT] *parameter_type*, ...]
IS
 [*procedure_declaration_section*]
BEGIN -- blocul procedurii
 procedure_executable_section
 [**EXCEPTION**
 procedure_exception_section]
END [*procedure_name*]; -- sfârșitul declarării procedurii

Funcții și proceduri utilizator - continuare

unde:

- ***procedure_name*** – este numele procedurii;
- ***parameter_name*** – reprezintă numele unui parametru formal din lista de parametri;
- ***parameter_type*** – reprezintă tipul unui parametru formal;
- ***procedure_declaration_section*** – reprezintă secțiunea de declarare de variabile pentru procedură;
- ***procedure_executable_section*** – reprezintă secțiunea executabilă a procedurii;
- ***procedure_exception_section*** – reprezintă secțiunea de tratare a excepțiilor din cadrul procedurii;
- ***block_variables*** - reprezintă secțiunea de declarare de variabile pentru blocul apelant;
- ***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 procedurii;

Definirea funcțiilor și procedurilor



Exemple:

- FUNCTION calcul_sold (v_data date) RETURN sold_init IS
v_sold_init_lei number := 0;
sold_init number;
BEGIN
 select nvl(sold, 0) into v_sold_init_lei from arh_contab.solduri
 where to_char(data, 'DD-MM-YYYY') = to_char(v_data, 'DD-MM-YYYY');
 sold_init:=v_sold_init_lei;
END;
- PROCEDURE completare_istoric(tip_op number) IS
operatie varchar2(12);
BEGIN
 if tip_op = 1 then operatie:='INSERARE';
 elsif tip_op = 2 then operatie:='STERGERE';
 lsif tip_op = 3 then operatie:='MODIFICARE';
 end if;

 insert into log.istoric(nume_tabel, utilizator, data_op, operatie, banca, entitate)
 values('BANCI', user, sysdate, operatie, :banci.cod_banca, :banci.cod_entitate);
END;

Apelul unei funcții/proceduri în Forms Builder

- Apelul funcției/procedurii poate fi făcut, de exemplu, prin activarea unui buton creat în formular, care apelează triggerul *WHEN-BUTTON-PRESSED* definit astfel:

- DECLARE

```
sold number:=0;
```

```
op number:=0;
```

```
BEGIN
```

```
    if :registru.data_sold is null then
```

```
        message('Precizati data soldului! ');
```

```
        raise form_trigger_failure;
```

```
    else
```

```
        sold:=calcul_sold(:registru.data_sold);
```

```
        op:=3;
```

```
    end if ;
```

```
    completare_istoric(op);
```

```
    .....
```

```
END;
```