



Universitatea
Politehnica
București



Facultatea de
Automatică și
Calculatoare



Catedra de
Calculatoare

Laborator 1

Blocuri de instrucțiuni PL/SQL

Autori:

Conf. Dr. Ing. Alexandru Boicea

As. Drd. Ing. Ciprian-Octavian Truică



Cuprins

- Introducere în PL/SQL
- Blocuri de instrucțiuni PL/SQL
 - Blocuri anonime
 - Blocuri înlanțuite
 - Blocuri imbricate
- Rularea blocurilor PL/SQL în SQL*Plus



Introducere în PL/SQL

- PL/SQL este un limbaj structurat pe blocuri
- Are caracteristici de procesare procedurală și de tratare a erorilor
- Blocurile pot fi de două tipuri:
 - Blocuri anonime
 - Subprograme(proceduri, funcții sau triggeri)



Blocuri anonime

- Blocul anonim are cea mai simplă structură și este compus din trei părți:
 - Declare – conține definirea tuturor variabilelor care vor fi folosite în procesul de execuție
 - Execuție – conține instrucțiuni DML și bucle de execuție
 - Excepții – conține rutinele de interceptare și tratare a erorilor



Blocuri anonime

- Fiecare bloc începe cu unul dintre cuvintele cheie ***declare*** sau ***begin***
- Cuvântul ***begin*** este punctul de început al blocului
- Înainte de executarea unui bloc se golește ***bufferul SQL***



- Structura logică a unui bloc PL/SQL anonim este:

DECLARE

- Secțiune pentru declararea de variabile

BEGIN

- Secțiune pentru instrucțiuni

EXCEPTION

- Secțiune pentru interceptarea și tratarea erorilor

END;



Blocuri anonime

- Ex. 1. Să se selecteze numele angajatului cu id-ul 7839.

```
set serveroutput on;
declare
    idemp number(4) := 7839;
    nume varchar(10);
begin
    select ename into nume from emp where empno = idemp;
    dbms_output.put_line('Numele angajatului este: ' || nume);
exception
    when no_data_found then
        dbms_output.put_line('Nu am gasit angajat cu id-ul ' || idemp);
end;
```



Blocuri anonime

- Obs. Pentru a afișa mesaje în SQL*Plus și SQL Developer se va folosi comanda:
`set serveroutput on;`
- Obs. Funcția care afișează mesaje în SQL*Plus și SQL Developer este:
`dbms_output.put_line(string);`
- Concatenarea șirurilor de caractere se face folosind operatorul `||`:
`'Numele angajatului este ' || nume`



Blocuri anonime

- Obs. Pentru a folosi select într-un bloc se va folosi o structură de forma:
 - SELECT COLOANA1, COLOANA2,... INTO
VARIABILA1, VARIABILA2... FROM TABEL1,...
 - Variabilele sunt declarate în secțiunea ***declare***
- Obs. Blocurile ***declare*** și ***exception*** nu sunt obligatorii.



Blocuri anonime

- Ex. 2. Să se insereze un nou angajat cu id-ul 1234, numele Georgescu, funcția Clerk, salariu 1250 și managerul cu id-ul 7566.

```
begin
```

```
    insert into emp(empno, ename, job, mgr, sal)  
        values(1234, 'Georgescu', 'clerk', 7566, 1250);
```

```
end;
```



Blocuri înlănțuite

- Blocurile anonime pot fi înlănțuite;
- Execuția se face secvențial;
- Declararea variabilelor pentru toate blocurile înlănțuite se face o singură dată în secțiunea *DECLARE*.



Blocuri înlănțuite

- Structura logică pentru un bloc înlănțuit este:

DECLARE – variabile bloc 1 și bloc 2

BEGIN

BEGIN – blocul 1

EXCEPTION – interceptarea și tratarea excepțiilor bloc 1

END; – sfârșit bloc 1

BEGIN – bloc 2

EXCEPTION – interceptarea și tratarea excepțiilor bloc 2

END; – sfârșit bloc 2

END;



Blocuri înlănțuite

- Ex. 3. Să se insereze un nou angajat cu id-ul 1235 introdus de la tastatura, numele Ionescu, funcția Analyst. După insert să se adauge pentru acest angajat un comision de 200.



Blocuri înlănțuite

declare

functie varchar2(9) := 'Analyst';

comision integer;

ecuson emp.empno%type := &empid;

nume emp.ename%type;

begin

begin

nume := 'Ionescu';

insert into emp(empno, ename, job)

values(ecuson, nume, upper(functie));

end;

begin

comision := 200;

update emp set comm = comision where empno = ecuson;

end;

end;



Blocuri înlănțuite

- Obs. Pentru a da o variabilă de la tastatură se folosește &;
- În exemplul de mai sus:
`ecuson emp.empno%type := &empid`
- Obs. Știind că variabila ecuson va trebui să fie de același tip precum coloana empno din tabelul emp, am folosit tipul de variabilă corelat `%type`.



Blocuri imbricate

- Blocurile imbricate sunt blocuri care se execută în interiorul altui bloc;
- Se mai numesc și sub-blocuri;
- Variabile pot fi transmise de la bloc la sub-bloc, **nu și invers.**



- Structura logică a unui bloc imbricat

DECLARE -- variabile bloc 1, bloc 2, bloc 3

BEGIN

BEGIN -- bloc 1

BEGIN -- bloc 2

BEGIN -- bloc 3

EXCEPTION -- tratare erori bloc 3

END; -- bloc 3

EXCEPTION -- tratare erori bloc 2

END; -- bloc 2

EXCEPTION -- tratare erori bloc 1

END; -- bloc 1

END;



- Ex. 4. Să se adauge un nou angajat, apoi să se modifice comisionul pentru acesta.



Blocuri imbricate

```
--rezolvare exercitiu 4
declare
  nume varchar2(17) := 'Populescu';
  functie varchar2(11) := 'Analyst';
begin
  begin

    insert into emp(empno, ename, job, sal)
      values(&ecuson, nume, functie, &salariu);
    /*
      acesta este un bloc imbricat
    */
    begin
      update emp set comm = &comision
        where empno =&ecuson and job=functie;
    end;
  end;
end;
```



Blocuri Imbricate

- Observatii:
- Comentariile pe mai multe linii se delimitează cu `/* */`
- Comentariile pe o singură linie se marchează cu două linii orizontale `--`
- Primul bloc face o inregistrare folosind variabile de substituie (**&&ecuson**);
- Al doilea bloc folosește variabila substituită *ecuson* (**&ecuson**) definită în primul bloc.



Rularea blocurilor PL/SQL în SQL*Plus

- Blocurile PL/SQL pot fi editate direct în linia de comandă sau pot fi editate în orice editor de text
- În SQL*Plus, editorul setat implicit se accesează tastând comanda **edit** (sau prescurtarea **ed**) sau folosind meniul *Editor* din bara de meniu
- Fișierele se salvează cu extensia .sql sau .txt



Rularea blocurilor PL/SQL în SQL*Plus

Comandă	Abreviere	Utilizare
APPEND text	A text	Adaugă text la sfârșitul liniei curente
CHANGE	C/old/new	Schimbă old text cu new text în linia curentă
CHANGE	C/text/	Șterge textul din linia curentă
CLEAR BUFFER	CL BUFF	Șterge toate liniile din bufferul SQL
DELETE	DEL	Șterge linia curentă
INPUT	I	Inserează un număr nedefinit de linii
INPUT text	I text	Inserează o linie constituită din text-ul respectiv
LIST	L	Listează toate liniile din buffer-ul SQL (ultimul bloc)
LIST n	L n	Listează o linie specificată de n
LIST m,n	L m,n	Listează toate liniile dinte m și n
RUN	R	Afișează și execută blocul curent din buffer
	/	Execută blocul curent din buffer



Rularea blocurilor PL/SQL în SQL*Plus

- Un bloc rămâne în buffer până la editarea următorului bloc
- Un bloc poate fi editat pe mai multe rânduri tastând ENTER la sfârșitul liniei curente și se poate executa astfel:
 - a. Se copiază blocul în editor (Copy/Paste) și se execută cu / sau RUN
 - b. Se execută tot scriptul (care poate să conțină mai multe blocuri) astfel:
 - SQL> @cale_fieier\nume_fisier.sql
 - SQL> start cale_fieier\nume_fisier.sql



Rularea blocurilor PL/SQL în SQL*Plus

- În linia de comandă se mai pot da următoarele comenzi:
 - SAVE cale_fișier\nume_fișier – permite salvarea într-un fișier a conținutului bufferului SQL
 - GET cale_fișier\nume_fișier – încarcă conținutul unui fișier salvat în prealabil în buffer
 - START cale_fișier\nume_fișier – lansează în execuție comanda salvată în fișier
 - RUN cale_fișier\nume_fișier – lansează în execuție comanda salvată în fișier
 - @cale_fișier\nume_fișier – lansează în execuție comanda salvată în fișier
 - EXIT – ieșire din SQL*Plus



Rularea blocurilor PL/SQL în SQL*Plus

Comandă	Utilizare
set serveroutput on;	Permite afișarea în SQL*Plus
set lines n	Permite printarea a n caractere pe verticală
set pages n	Permite printarea a n coloane



Exemple

- Pentru următoarele exerciții se va crea încă un tabel numit mesaje:

```
create table mesaje  
{  
    mesaj1 varchar2(40) ,  
    mesaj2 varchar2(40)  
};
```



Exemple

- Ex. 5. În exemplul următor se prezintă un bloc PL/SQL care face o verificare a numărului de angajați din anul 1992. Dacă nu există nicio angajare, se va declanșa o excepție **no_data_found** și se va insera în tabela **mesaje**, mesajul numărul 1, iar dacă sunt mai mulți angajați, se va declanșa excepția; **too_many_rows** și se va insera mesajul numărul 2.



Exemple

```
declare
  v_nume varchar2(20);
  v_job  varchar2(20);
begin
  select ename, job into v_nume, v_job from emp
    where hiredate between '1-jan-1992' and '31-dec-1992';
exception
  when no_data_found then
    insert into mesaje values(1, 'Nu exista angajari in anul 1992');
  when too_many_rows then
    insert into mesaje values(2, 'Sunt mai multe angajari in anul 1992');
end;
```

- Pentru a vedea rezultatul procedurii se va face un select pe tabela mesaje:

```
SQL> SELECT * FROM MESAJE;
```



Exemple

- Ex. 6. În exemplul de mai jos se folosesc două blocuri imbricate care afișează numele, data angajării și comisionul unui angajat introducând id-ul angajatului și id-ul departamentului de la tastatură.

```
set serveroutput on;  
declare  
    idDepartament number(4);  
    idAngajat number(4);  
    numeAngajat varchar(30);  
    comision number(10);  
    dataAngajare date;
```



Exemple

```
begin
  idDepartament := &idDept;
  idAngajat := &idAng;
begin
  select ename, hiredate, comm into numeAngajat, dataAngajare, comision
    from emp where empno = idAngajat and deptno=idDepartament;
  dbms_output.put_line(rpad('Nume',30,' ')||
    rpad('Data Angajare',15,' ')||lpad('Comision', 10,' '));
  dbms_output.put_line(rpad(numeAngajat,30,' ')||
    rpad(dataAngajare,15,' ')||lpad(comision, 10,' '));
exception
  when no_data_found then null;
end;
end;
```



Exemple

- Ex. 7. În următorul exercițiu vom edita un fișier ex7.sql care conține un bloc PL/SQL.

```
set serveroutput on;
```

```
set lines 150;
```

```
set pages 100;
```

```
declare
```

```
    numeDepartament varchar2(20);
```

```
    salariu number;
```



Exemple

```
begin
  select sum(sal+nvl(comm,0)) into salariu from emp
    where deptno=&idDept and job='&job' group by deptno;
  select dname into numeDepartament from dept
    where deptno=&idDept;
  insert into mesaje values('Suma veniturilor in departamentul '
    ||&idDept, 'este '||to_char(salariu));
  exception
    when no_data_found then
      insert into mesaje(mesaj1) values('In departamentul '
        ||&idDept||' nu exista angajati');
end;
/

undef idDept;

select * from mesaje;
```




Exemple

- Executarea fișierului:

```
SQL>@cale_fisier\ex7.sql
```

- Să se testeze pentru funcția ANALYST și departamentele 20, 40, 50.