

Baze de Date II

(Proiectarea Sistemelor cu Baze de Date)

Cornelia TUDORIE



Proiectarea Sistemelor cu Baze de Date - Cuprins

- A. Modelul Relațional.**
- B. Limbajul SQL. Lucrul cu tabele.**
- C. Sisteme cu baze de date. Performanța.**
- D. Obiecte ale sistemului de baze de date. Performanța în utilizare**
- E. Programe pentru baze de date. Performanța în programare.**
- F. Protecția datelor.**
- G. Sisteme Informatică. Proiectarea sistemelor de baze de date.**

E. Programe pentru baze de date. Performanța în programare.

- 1. Fișiere de comenzi**
- 2. PL/SQL. Tipuri de programe**
- 3. Blocuri anonime**
- 4. Subprograme stocate. Pachete**
- 5. Declanșatori**
- 6. Tranzactii**

E. 3. Blocuri anonime

- Bazele limbajului PL/SQL
- Accesul la baza de date
- **Structuri de control**
- Tipuri de date compuse
- Cursori expliți
- Excepții

Structuri de control.

SQL - limbaj non procedural (declarativ)

PL/SQL - limbaj procedural



structuri de control in PL/SQL

Structuri de control.

1. *Structuri alternative*
2. *Bucle*

Structuri de control.

Structuri alternative

IF - THEN

IF - THEN - ELSE

IF - THEN - ELSIF - ELSE

CASE

Structuri de control.

Structura IF - THEN

IF *conditie* THEN

instructiuni executabile ;

END IF ;

Structuri de control.

Structura alternativa **IF - THEN**

Exemple

```
declare
v_nume emp.ename%TYPE := &nume;
v_sal emp.sal%TYPE;
begin
    SELECT sal INTO v_sal
    FROM emp
    WHERE ename = v_nume;
    IF v_sal > 2500
    THEN
        dbms_output.put_line (' angajatul ' || v_nume || ' are un salariu bun ');
    END IF ;
end;
```

Structuri de control.

Structura alternativa **IF - THEN - ELSE**

```
IF conditie THEN  
    instructiuni_pentru_true ;  
[ ELSE  
    instructiuni_pentru_false ; ]  
END IF ;
```

Structuri de control.

Structura alternativa

IF - THEN - ELSE

Exemple

```
declare
v_num emp.ename%TYPE := &nume;
v_comm emp.comm%TYPE;
begin
    SELECT comm INTO v_comm
    FROM emp
    WHERE ename = v_num;
    IF v_comm IS NOT NULL
    THEN
        dbms_output.put_line (' angajatul ' || v_num || ' primeste comision ');
    ELSE
        dbms_output.put_line (' angajatul ' || v_num || ' nu primeste comision ');
    END IF ;
end;
```

Structuri de control.

Structura alternativa

IF - THEN - ELSIF - ELSE

```
IF conditie_1 THEN  
    instructiuni_1 ;  
!!!! [ ELSIF conditie_n THEN  
        instructiuni_n ; ]  
[ ELSE  
    instructiuni ; ]  
END IF ; !!!!
```


Structuri de control.

Structura alternativa

IF - THEN - ELSIF - ELSE

Exemple

```
declare val NUMBER :=&valoare;
begin
    IF val <10 THEN
        dbms_output.put_line (' valoare inferioara lui 10 ');
    ELSIF val <100 THEN
        dbms_output.put_line (' valoare intre 10 si 99 ');
    ELSIF val <=500 THEN
        dbms_output.put_line (' valoare intre 100 si 500 ');
    ELSE
        dbms_output.put_line (' valoare prea mare ');
    END IF ;
end;
```


Structuri de control.

Structura alternativa IF - THEN - ELSE

Exemplu - IF imbricat

```
declare val NUMBER :=&valoare;
begin
    IF val <10 THEN
        dbms_output.put_line (' valoare inferioara lui 10 ');
    ELSE
        IF val <100 THEN
            dbms_output.put_line (' valoare intre 10 si 99 ');
        ELSE
            IF val <=500 THEN
                dbms_output.put_line (' valoare intre 100 si 500 ');
            ELSE
                dbms_output.put_line (' valoare prea mare ');
            END IF ;
        END IF ;
    END IF ;
end;
```

Structuri de control.

Structura alternativa

CASE simplu

CASE *selector*

WHEN *expr_1* **THEN**

instructiuni_1 ;

[**WHEN** *expr_2* **THEN**

instructiuni_2 ; ...]

[**ELSE**

instructiuni ;]

!!!!

END CASE ;

Structuri de control.

Structura alternativa

CASE simplu

Exemple

```
declare
    val NUMBER := &valoare;
begin
    case val
        when 1 then
            dbms_output.put_line (' ati introdus valoarea 1 ');
        when 2 then
            dbms_output.put_line (' ati introdus valoarea 2 ');
        when 3 then
            dbms_output.put_line (' ati introdus valoarea 3 ');
        else
            dbms_output.put_line (' alta valoare ');
        end case;
    end;
```

Structuri de control.

Structura alternativa

CASE simplu

Exemple

```
declare
    val NUMBER := &valoare;
begin
    case val
        when 1 then
            dbms_output.put_line (' ati introdus valoarea 1 ');
        when 2 then
            dbms_output.put_line (' ati introdus valoarea 2 ');
        when 3 then
            dbms_output.put_line (' ati introdus valoarea 3 ');
        else
            NULL;
    end case;
end;
```


Structuri de control.

Structura alternativa

CASE multiplu

CASE

WHEN *conditie_1* **THEN**

instructiuni_1 ;

[**WHEN** *conditie_2* **THEN**

instructiuni_2 ; ...]

[**ELSE**

instructiuni ;]

END CASE ;

Structuri de control.

Structura alternativa

CASE multiplu

Exemple

```
declare
    val NUMBER := &valoare;
begin
    case
        when val <10 then
            dbms_output.put_line (' valoare inferioara lui 10 ');
        when val <100 then
            dbms_output.put_line (' valoare entre 10 si 99 ');
        when val <=500 then
            dbms_output.put_line (' valoare intre 100 si 500 ');
        else
            dbms_output.put_line (' valoare prea mare ');
        end case ;
    end;
```

Structuri de control.

Structura alternativa

CASE multiplu

Exemplu - conditiile sunt independente!

```
declare v NUMBER := 168 ;
begin
  case
    when MOD(v,5)=0 then
      dbms_output.put_line (' valoarea este divizibila prin 5 ');
    when v BETWEEN 100 and 999 then
      dbms_output.put_line (' valoare intre 100 si 999 ');
    when v=168 then
      dbms_output.put_line (' valoarea este 168 ');
    when 5 < 7 then
      dbms_output.put_line (' 5 < 7 ');
  end case ;
end;
```

Structuri de control.

Bucle

structuri: bucla simpla
FOR
WHILE
UNTIL

Structuri de control.

Bucła simpla

LOOP *conditie* **THEN**

...

instructiuni executabile ;

...

[**EXIT** [**WHEN** *conditie*] ;]

...

END LOOP ;

Structuri de control.

Bucla **FOR**

```
FOR contor IN [ REVERSE ] val_inf.. val_sup  
LOOP  
    instructiuni executabile ;  
END LOOP ;
```


Structuri de control.

*Buc*la FOR

Exemplu

```
declare v number :=168;  
begin  
  dbms_output.put_line (' divizorii lui ' || v || ' mai mici ca 10 ');  
  for i in 1..9  
    loop  
      if MOD( v , i )=0 then  
        dbms_output.put_line (' valoarea este divizibila prin ' || i );  
      end if;  
    end loop ;  
end;
```

variabila declarata implicit;
ea nu este valabila decat
in interiorul buclei!

Structuri de control.

Bucla **FOR**

Exemplu

```
begin
    dbms_output.put_line ( ' numaratoare inversa : ' ) ;
for i in reverse 1..10
    loop
        dbms_output.put_line ( i ) ;
    end loop ;
end;
```

Structuri de control.

Bucla **WHILE** (*cat timp*)

WHILE *conditie*

LOOP

instructiuni executabile ;

END LOOP ;

Structuri de control.

Bucla WHILE

Exemplu

declare

```
q number (2) := 0;  
a number := &val_a;  
b number := &val_b;  
r number ;
```

begin

```
dbms_output.put_line ( ' Impartire prin scadere : ' ) ;  
r:=a;  
while r >= b  
loop  
    r := r-b; q:=q+1;  
end loop ;  
dbms_output.put_line ( a || ' impartit la ' || b || ' este ' || q || ' rest ' || r);
```

end;

Structuri de control.

Bucla **UNTIL** (*pana cand*)

LOOP

instructiuni executabile ;

EXIT WHEN *conditie;*

END LOOP ;

Structuri de control.

Bucula UNTIL

Exemplu

```
declare
    q number (2) := 0;
    a number := &val_a;
    b number := &val_b;
    r number ;
begin
    dbms_output.put_line ( ' Impartire prin scadere : ' ) ;
    r:=a;
    if r >= b then
        loop
            r := r-b; q:=q+1;
            exit when r < b;
        end loop ;
    end if;
    dbms_output.put_line ( a || ' impartit la ' || b || ' este ' || q || ' rest ' || r);
end;
```

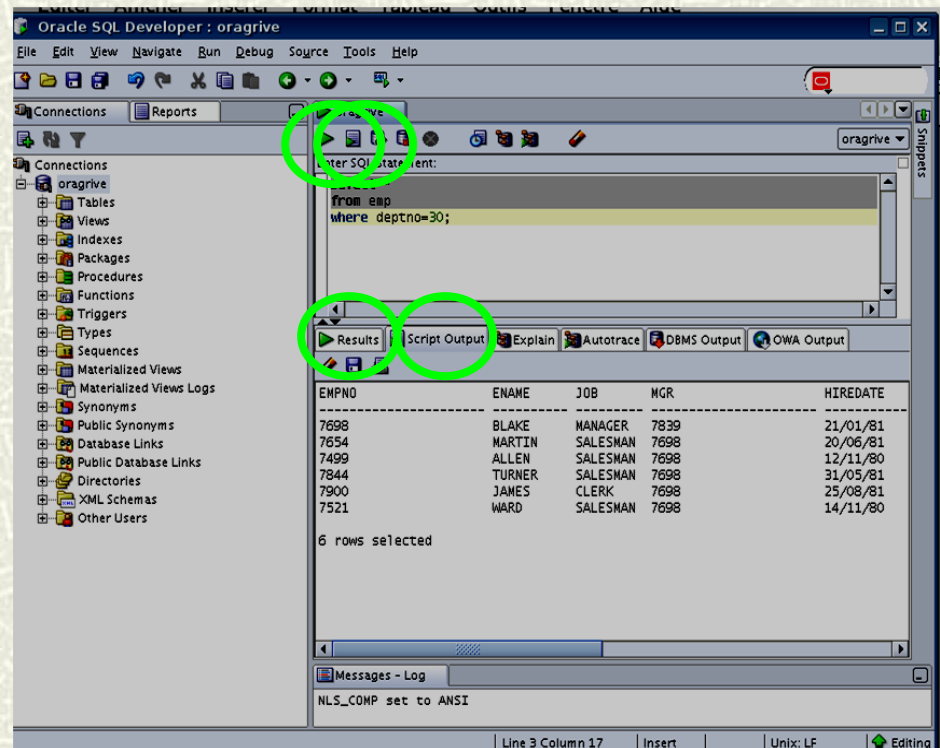
PL/SQL la laborator

Interfata utilizata la laborator:
SQLDeveloper

Comenzi SQL executate:
individual sau
ca fisiere de comenzi

Rezultat ->

Results sau
Script Output

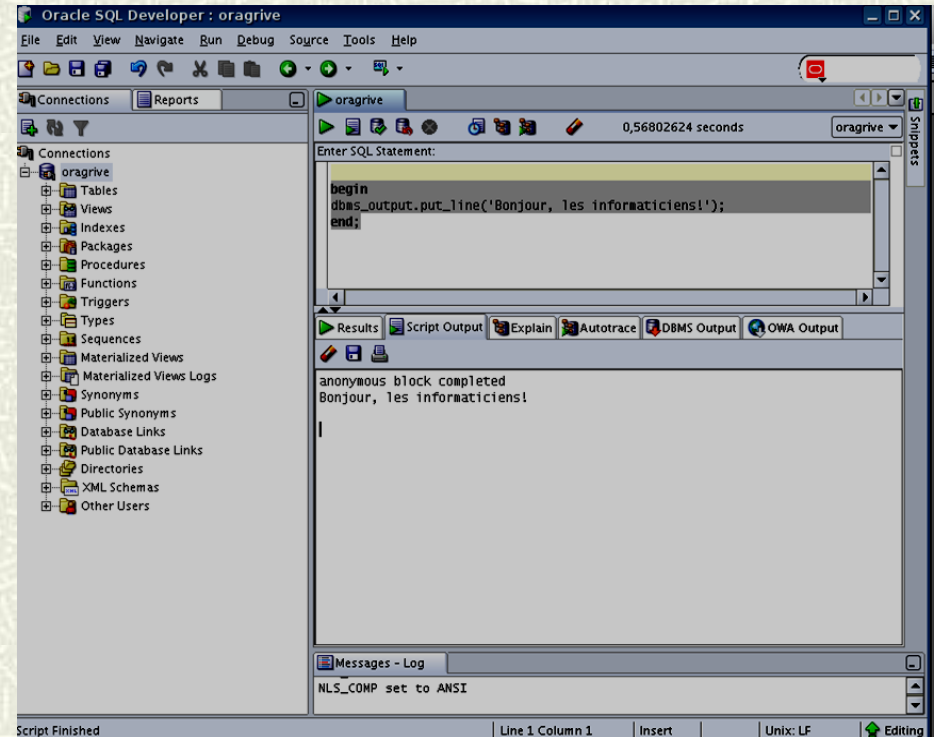


PL/SQL la laborator

Interfata utilizata la laborator:
SQLDeveloper

Blocuri SQL executate:
ca **fisiere de comenzi**

Rezultat ->
Script Output



PL/SQL la laborator

Interfata utilizata la laborator:
SQLPlus

```
SQL>
SQL>
SQL> begin
  2  dbms_output.put_line('Bonjour, les informaticiens!');
  3  end;
  4  /

PL/SQL procedure successfully completed.

SQL> █
```


PL/SQL la laborator

Interfata utilizata la laborator:

SQLPlus : blocuri PL/SQL scrise direct in buffer

```
SQL> begin  
2 dbms_output.put_line('Buna ziua, studenti!');  
3 end;  
4 /
```

PL/SQL procedure successfully completed.

```
SQL> |  
1 begin  
2 dbms_output.put_line('Buna ziua, studenti!');  
3* end;  
SQL> /
```

PL/SQL procedure successfully completed.

SQL>

PL/SQL la laborator

Interfata utilizata la laborator:

SQLPlus : blocuri PL/SQL scrise in fisiere de comenzi

Fisierul text bloc1.sql:

```
begin
  dbms_output.put_line('Buna ziua, studenti!');
end;
/
```

Executie:

SQL> @bloc1

PL/SQL procedure successfully completed.

SQL>