

Tema 2 – PL/SQL

1. Ce va afișa acest bloc?

Declare

```
fname varchar2(20);
```

```
lname varchar2(15) default 'fernandez';
```

Begin

```
DBMS_OUTPUT.PUT_LINE(fname|| ' ' || lname);
```

End;

- a. Va printa 'fernandez'.
- b. Va returna o eroare deoarece variabila fname este utilizată fără a fi inițializată.
- c. Se execută cu succes și va printa "null fernandez".
- d. Va returna o eroare deoarece nu se poate folosi cuvântul cheie DEFAULT pentru a inițializa o variabilă de tip varchar2.
- e. Va returna o eroare deoarece variabila fname nu este declarată.

A

2. În Application Express rulați exemplul de mai jos pentru a vedea rezultatul obținut:

Create or replace function num_characters (p_string IN VARCHAR2) return integer as

```
v_num_characters INTEGER;  
begin  
  select length(p_string) into v_num_characters from dual;  
  return v_num_characters;  
end;  
  
declare  
  v_length_of_string INTEGER;  
begin  
  v_length_of_string := num_characters('Oracle Corporation');  
  DBMS_OUTPUT.PUT_LINE(v_length_of_string);  
end;
```

Se poate schimba numele funcției cu altul. Va afișa 18.

3.

a. Creați o tabelă numită emp_dup care să fie duplicată a tabelii employees.

```
Create table emp_dup as select * from employees
```

b. Selectați câmpurile first_name și last_name din tabelă emp_dup.

```
select first_name , last_name from emp_dup
```

c. Rulați codul de jos:

```
Declare  
last_name varchar2(25) := 'King';  
Begin  
update emp_dup set first_name = 'Jennifer' where last_name = last_name;  
End;
```

Ce credeți că se va întâmpla?

Toti sunt Jennifer.

Secvența verifică dacă valoarea din last_name (coloana din tabelă) este egală cu last_name (aceeași coloană din tabelă) și înlocuiește first_name cu Jennifer.

d. Selectați din nou câmpurile first_name și last_name din tabelă emp_dup pentru a vă confirma ipoteza.

```
select first_name , last_name from emp_dup
```

e. Modificați codul de mai sus pentru a fi modificate prenumele celor al căror nume este Chen. Pentru aceasta eliminați întâi tabelă emp_dup și recreați-o.

```
drop table emp_dup
```

```
declare
```

```
v_last_name varchar2(25) := 'King';
```

Begin

```
update emp_dup set first_name = 'Jennifer' where last_name = v_last_name;
```

End;

4. Care din variabilele de mai jos sunt valide? Explicați de ce.

number_of_students PLS_INTEGER.; - da

student_name VARCHAR2(10):=Popescu; - nu e valid, trebuie ‘

stu_per_class constatnt number; - nu e valid, trebuie valoare

today date := sysdate+1; - da

Scrieți un bloc anonim PL/SQL în care să declarați și să afișați fiecare variabilă de mai sus, corectând declarațiile invalide.

declare

```
number_of_students PLS_INTEGER:=3;
```

```
student_name varchar2(10):='Popescu';
```

```
stu_per_class constant number:=30;
```

```
today date:=sysdate+1;
```

begin

```
dbms_output.put_line(number_of_students||' '||student_name||' '||stu_per_class||' '||today);
```

end;

5. Examinați variabilele din codul de mai jos și spuneți dacă au fost denumite corect ? Dacă nu, de ce nu ? Dacă nu au fost denumite corect atunci care ar fi un nume bun pentru ele și de ce?

Declare

Country_name varchar2(50);

Median_age number(6,2);

Begin

```
Select country_name, median_age into country_name, median_age
From wf_countries
Where country_name = 'United States of America';

DBMS_OUTPUT.PUT_LINE('Median age in ' || country_name || ' is '
||median_age||'.');

End;
```

Schimbați declarațiile variabilelor astfel încât să folosiți atributul %TYPE.

Nu e bine sa folositi nume pentru variabile identice cu nume de câmpuri din tabele. Vezi ca exemplu exercițiul 3.e.

```
v_median_age wf_countries.median_age%TYPE
```

6. Examinați următorul cod și spuneți care credeți că va fi rezultatul afișat? De ce? Ce se întâmplă?

Declare

```
X varchar2(20);
```

Begin

```
X:= '123' + '456';
```

```
DBMS_OUTPUT.PUT_LINE(x);
```

End;

Se face conversie implicita.

7. Creați următorul bloc anonim:

Begin

```
DBMS_OUTPUT.PUT_LINE('Hello World');
```

End;

Adăugați o secțiune declarativă acestui bloc PL/SQL unde declarați următoarele variabile:

- o variabilă today de tip DATE pe care o inițializați cu sysdate.
- o variabilă tomorrow de tipul variabilei today. Folosiți %TYPE.

În secțiunea declarativă inițializați variabila tomorrow cu o expresie care calculează ziua de mâine. Afișați valoarea lui today și tomorrow după ce afișați 'Hello World'.

Declare

```
today date:=sysdate;  
tomorrow today%type:=today+1;
```

Begin

```
Dbms_output.put_line('Hello World'|| today||' '||tomorrow);
```

End;

8. Scrieți un bloc PL/SQL anonim în care să declarați o variabilă my_date având ca valoare ziua curentă și afișați-o sub forma 'Month dd, yyyy'. Creați o altă variabilă numită v_last_day de tip date la care atribuiți ultima zi din lună și afișați valoarea.

```
my_date date:=sysdate;
```

```
v_last_day date;
```

Begin

```
DBMS_OUTPUT.PUT_LINE(to_char(sysdate, 'Month dd, yyyy'));
```

```
v_last_day:=last_day(my_date);
```

```
DBMS_OUTPUR.PUT_LINE(v_last_day);
```

End;

9. Rulați codul de mai jos. Explicați ce se întâmplă.

Declare

```
V_number number(2);
```

```
Begin
    V_number := 9999;
Exception
    When others then
        DBMS_OUTPUT.PUT_LINE('Exceptie');
End;
```

Ștergeți secțiunea de excepții și rulați codul. Explicați rezultatul.

Valoare numar prea mare.

10. Care este valoarea lui v_boolean?

```
Declare
    v_number number;
    v_boolean boolean;
Begin
    v_number := 25;
    v_boolean := NOT(v_number > 30);
END;

True
```

11. Rulați codul următor și explicați rezultatul.

```
Declare
    v_number number(4) ;
Begin
    v_number :=1234;

    Declare
        v_number number(4);
    Begin
        v_number :=5678;
        v_number := 'un string';
    End;

Exception
    When others then
        DBMS_OUTPUT.PUT_LINE('o exceptie are loc');
        DBMS_OUTPUT.PUT_LINE('numarul este '||v_number);
```

End;

Exceptia nu este tratata in blocul interior ea fiind propagata in cel exterior. Valoare numar: 1234.

12. Creați o copie a tabelii departments numita new_depts. Verificați structura acestei tabele. Rulați apoi codul:

Declare

```
v_max_deptno new_depts.department_id%type;
```

Begin

```
Select max(department_id) into v_max_deptno
```

```
From new_depts;
```

```
DBMS_OUTPUT.PUT_LINE('Ultimul departament este '|| v_max_deptno );
```

End;

Adăugați două noi variabile:

```
v_dept_name new_depts.department_name%TYPE := 'Un nou departament';
```

```
v_dept_id new_depts.department_id%TYPE;
```

Modificați codul astfel încât id-ul noului departament: v_dept_id să ia valoarea id-ul maxim (department_id) adica ce am pus in variabila v_max_deptno, plus 10. Adăugați o declarație INSERT astfel încât să inserați un nou rând în tabela new_depts folosind v_dept_id și v_dept_name pentru a popula id-ul și numele departamentului iar pentru location_id și manager_id folosiți NULL.

```
v_dept_id:=10+v_max_deptno;
```

```
Insert into new_depts values(v_dept_id, v_dept_new, null, null);
```

13. Examinați blocul de jos. Ce credeți că va fi salvat în baza de date? Rulați codul pentru a vă testa ipoteza.

BEGIN

```
Insert into students values (100, 'VASILE', 'BUGET');
```

```
SAVEPOINT sp_100;
```

```
Insert into stundets values (101, 'MARIA', 'TAXA');
```

```
Insert into students values (102, 'IONUT', 'BUGET');  
ROLLBACK TO sp_100;  
COMMIT;  
END;
```

Doar primul insert.

14. Examinați blocul de jos. Ce credeți că va fi salvat în baza de date? Rulați codul pentru a vă testa ipoteza.

```
BEGIN  
Insert into students values (100, 'VASILE', 'BUGET');  
SAVEPOINT sp_400;  
Insert into students values (101, 'MARIA', 'TAXA');  
ROLLBACK; -- rollback simplu anuleaza tot inclusiv savepoint-ul  
Insert into students values (102, 'IONUT', 'BUGET');  
ROLLBACK TO sp_400;  
COMMIT;  
END;
```

Eroare nu găsește sp_400.

15. Creați o tabela numită EVALUARE cu câmpurile: nume_student, nume_curs, nota, data și populați-o. Ipotetic, dacă studentul Ionescu are nota la cursul SGDB între 5 și/ sau 7 afișați acest lucru, dacă are între 7 și/ sau 9 atunci afișați acest lucru și dacă are peste 9 atunci afișați un mesaj corespunzător. Folosiți CASE.

```
Create table evaluare (nume_student varchar(30), nume_curs varchar(30), nota  
INTEGER, data DATE )
```

```
insert into evaluare values('Ionescu', 'sgbd', 10, sysdate);
```

```
Declare
```

```
V_nota evaluare.nota%type;
```

```
Begin
```

```
    Select nota into v_nota from evaluare where nume_student='Ionescu';
```

```
Case
```

```
When v_nota >5 and v_nota <=7 then DBMS_OUTPUT.PUT_LINE('intre 5 si 7');
```

```
When v_nota >7 and v_nota <=9 then DBMS_OUTPUT.PUT_LINE('intre 7 si 9');
```

```
When v_nota >9 then DBMS_OUTPUT.PUT_LINE('peste 9');
```


End case;
End;

16.

declare

cursor distributie_zodie is

```
select nume_zodie, count(*) distributie from zodiac z join utilizatori u on
to_date(to_char(u.data_nastere, 'DD-MM'), 'DD-MM') between to_date(z.data_inceput,
'DD-MM')
and to_date(z.data_sfarsit, 'DD-MM')
group by nume_zodie;
```

cursor stud(zodie varchar2) is select nume, prenume, data_nastere from utilizatori u join zodiac z on

```
to_date(to_char(u.data_nastere, 'DD-MM'), 'DD-MM') between to_date(z.data_inceput,
'DD-MM') and
```

```
to_date(z.data_sfarsit, 'DD-MM')
```

```
where z.nume_zodie = zodie;
```

begin

```
for v_zodie in distributie_zodie loop
```

```
dbms_output.put_line(v_zodie.nume_zodie||' '||v_zodie.distributie);
```

```
for rand in stud(v_zodie.nume_zodie) LOOP
```

```
dbms_output.put_line(rand.nume||' '||rand.prenume||' '||rand.data_nastere );
```

```
end loop;
```

```
end loop;
```

```
end;
```