

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;
```

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

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

```
create 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;
```

3.

- Creați o tabelă numită emp_dup care să fie duplicată a tabelii employees.
- Selectați câmpurile first_name și last_name din tabelă emp_dup.
- Rulați codul de jos:

```
Declare
    last_name varchar2(25) := 'Chen';
Begin
```

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

Ce credeți că se va întâmpla?

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

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

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

```
number_of_students PLS_INTEGER.;  
student_name       VARCHAR2(10):=Popescu;  
stu_per_class      constatnt number;  
today              date := sysdate+1;
```

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

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.

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;
```

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'.

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.

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.

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;
```

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;
```

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('Cel mai nou departament este '|| v_new_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 (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.

13. Creați o tabela noua cu denumirea aleasă de voi, care să aibă trei câmpuri: pentru cod student, nume student si forma de finanțare.

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

```
BEGIN
Insert into <nume tabelă> values (100, 'VASILE', 'BUGET');
SAVEPOINT sp_100;
```

```
Insert into <nume tabelă> values (101,'MARIA', 'TAXA');  
Insert into <nume tabelă> values (102, 'IONUT', 'BUGET');  
ROLLBACK TO sp_100;  
COMMIT;  
END;
```

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

```
BEGIN  
Insert into <nume tabelă> values (100, 'VASILE', 'BUGET');  
SAVEPOINT sp_400;  
Insert into <nume tabelă> values (101,'MARIA', 'TAXA');  
ROLLBACK;  
Insert into <nume tabelă> values (102, 'IONUT', 'BUGET');  
ROLLBACK TO sp_400;  
COMMIT;  
END;
```

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 7 afișați acest lucru, dacă are între 7 și 9 atunci afișați acest lucru și dacă are peste 9 atunci afișați un mesaj corespunzător. Folosiți CASE.