

Laborator 4:

STRUCTURI FUNDAMENTALE DE PROGRAMARE

(partea II)

2. STRUCTURI REPETITIVE

Structura **LOOP.....END LOOP**

```
LOOP  
    Secventa comenzi;  
    EXIT [WHEN cond];  
END LOOP;
```

Exemplul 1:

Se afișează pe ecran utilizând structura **LOOP...END LOOP** numerele 9,7, 4, 0.

```
DECLARE  
    v_nr number(2):=10;  
    i number(2):=1;  
BEGIN  
    loop  
        v_nr:=v_nr-i;  
        i:=i+1;  
        exit when v_nr < 0;  
        dbms_output.put_line(v_nr);  
    end loop;  
END;
```

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation tabs: APEX, App Builder, SQL Workshop, Team Development, and Gallery. Below the tabs, there's a section for "SQL Commands" with a "Language" dropdown set to "PL/SQL", "Rows" set to "10", and buttons for "Clear Command" and "Find Tables". The main area contains a PL/SQL script:

```
1 DECLARE
2     v_nr number(2):=10;
3     i number(2):=1;
4 BEGIN
5     loop
6         v_nr:=v_nr-i;
7         i:=i+1;
8         exit when v_nr < 0;
9         dbms_output.put_line(v_nr);
10    end loop;
11 END;
```

Below the script, there are tabs for "Results", "Explain", "Describe", "Saved SQL", and "History". The "Results" tab is active, showing the output of the script:

```
9
7
4
0
```

Below the output, it says "Statement processed." and "0.01 seconds".

Exemplul 2:

Se afișează în ordine angajații care au salariul mai mic decat salriul mediu din firma.

```
DECLARE
    v_sal emp.sal%type;
    v_salmediu emp.sal%type;
BEGIN
    SELECT avg(sal)
    INTO v_salmediu
    FROM emp;
    dbms_output.put_line('Salariul mediu este: '||v_salmediu);

    FOR c IN ( SELECT EMPNO, ENAME, SAL
                FROM emp
                WHERE deptno = 10 )
    LOOP
        IF(v_salmediu < c.sal) THEN
            DBMS_OUTPUT.PUT_LINE ('Salariul pentru angajatul ' || c.ename || '
este: ' || c.sal);
        END IF;
    END LOOP;
END;
```

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation tabs: APEX, App Builder, SQL Workshop, Team Development, and Gallery. Below this is the 'SQL Commands' section with a language dropdown set to 'SQL' and a 'Rows' limit of 10. The main area contains a PL/SQL script with line numbers 1 through 19. The script declares two variables, calculates an average salary, and then loops through employees in department 10 to compare their salary to the average. The results pane at the bottom shows the output of the script.

```

1 DECLARE
2     v_sal emp.sal%type;
3     v_salmediu emp.sal%type;
4 BEGIN
5     SELECT avg(sal)
6     INTO v_salmediu
7     FROM emp;
8     dbms_output.put_line('Salariul mediu este: '||v_salmediu);
9
10    FOR c IN ( SELECT EMPNO, ENAME, SAL
11              FROM emp
12              WHERE deptno = 10 )
13    LOOP
14        IF(v_salmediu < c.sal) THEN
15            DBMS_OUTPUT.PUT_LINE ('Salariul pentru angajatul ' || c.ename || ' este: ' || c.sal);
16        END IF;
17    END LOOP;
18 END;
19

```

Results pane output:

```

Salariul mediu este: 2180.18
Salariul pentru angajatul KING este: 5000
Salariul pentru angajatul CLARK este: 2450
Statement processed.

```

Structura WHILE.....LOOP....END LOOP

WHILE conditie **LOOP**
Secventa comenzi 1;
Secventa comenzi 2;
EXIT [WHEN conditie];
END LOOP;

Exemplul 3:

Se afișează pe ecran utilizând structura **WHILE LOOP...END LOOP** numerele 9,7, 4, 0.

```

DECLARE
    v_nr number(2):=10;
    i number(2):=1;
BEGIN
    while v_nr > 0 loop
        v_nr:=v_nr-i;
        i:=i+1;
        dbms_output.put_line(v_nr);
    end loop;
END;

```

The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation links for APEX, App Builder, SQL Workshop, Team Development, and Gallery. Below this is a 'SQL Commands' section with a 'Language' dropdown set to 'PL/SQL' and 'Rows' set to '10'. There are buttons for 'Clear Command' and 'Find Tables'. The main area contains a PL/SQL script:

```
1 DECLARE
2     v_nr number(2):=10;
3     i number(2):=1;
4 BEGIN
5     while v_nr > 0 loop
6         v_nr:=v_nr-i;
7         i:=i+1;
8         dbms_output.put_line(v_nr);
9     end loop;
10 END;
11
```

Below the script, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing the output of the script:

```
9
7
4
0

Statement processed.

0.01 seconds
```

Exemplul 4:

Sa se afiseze suma cifrelor unui numar intreg.

```
DECLARE
  n INTEGER;
  temp_sum INTEGER;
  r INTEGER;
BEGIN
  n := 123456;
  temp_sum := 0;
  WHILE n <> 0 LOOP
    r := MOD(n, 10);
    temp_sum := temp_sum + r;
    n := Trunc(n / 10);
  END LOOP;
  dbms_output.Put_line('sum of digits = ' || temp_sum);
END;
```

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language PL/SQL Rows 10 Clear Command Find Tables

```

1 DECLARE
2     n INTEGER;
3     temp_sum INTEGER;
4     r INTEGER;
5 BEGIN
6     n := 123456;
7     temp_sum := 0;
8
9     WHILE n <> 0 LOOP
10        r := MOD(n, 10);
11        temp_sum := temp_sum + r;
12        n := Trunc(n / 10);
13    END LOOP;
14    dbms_output.Put_line('sum of digits = ' || temp_sum);
15 END;
16

```

Results Explain Describe Saved SQL History

sum of digits = 21

Statement processed.

0.01 seconds

Structura FOR.....LOOP....END LOOP

FOR var IN [REVERSE] valmin..valmax LOOP
Secventa comenzi;
EXIT [WHEN conditie];
END LOOP;

Observatii:

Valorile intervalului pot fi de orice tip, dar sa aiba valori care pot fi convertite la un intreg (de exemplu 20/13 sau 11/5). Daca aceste 2 valori vor fi egale ca intregi atunci instructiunile din interiorul ciclului se executa o singura data. De exemplu secventa:

```

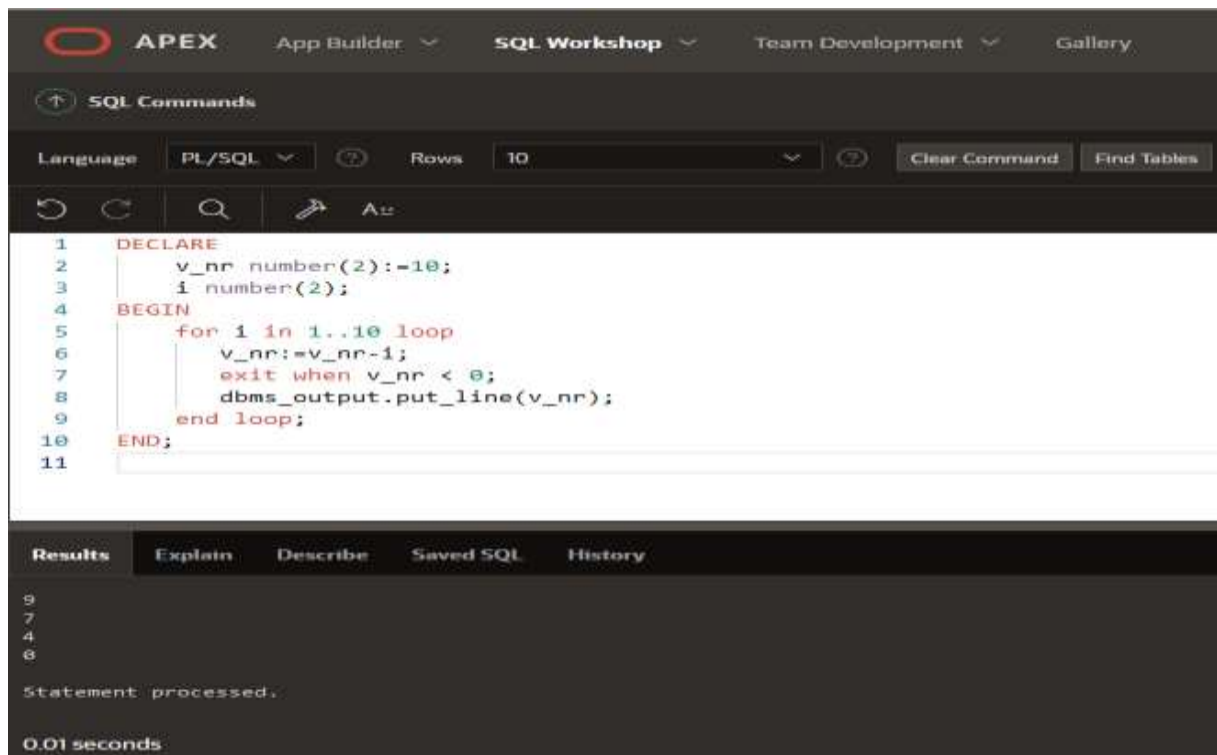
FOR i IN 3..3 LOOP
    Secventa comenzi;
END LOOP;

```

Exemplul 5:

Se afișează pe ecran utilizând structura **FOR...END LOOP** numerele 9,7, 4, 0.

```
DECLARE
  v_nr number(2):=10;
  i number(2);
BEGIN
  for i in 1..10 loop
    v_nr:=v_nr-i;
    exit when v_nr < 0;
    dbms_output.put_line(v_nr);
  end loop;
END;
```



```
1 DECLARE
2   v_nr number(2):=10;
3   i number(2);
4 BEGIN
5   for i in 1..10 loop
6     v_nr:=v_nr-i;
7     exit when v_nr < 0;
8     dbms_output.put_line(v_nr);
9   end loop;
10 END;
11
```

Results

```
9
7
4
0
```

Statement processed.

0.01 seconds

Exemplul 6:

Se afișează în ordine angajații cu codurile în intervalul 100-110 atât timp cât salariul acestora este mai mic decât media:

```
DECLARE
  v_sal angajati.salariul%type;
  v_salMediu v_sal%type;
  -- i nu mai trebuie declarat
BEGIN
  SELECT avg(salariul) into v_salmediu from angajati;
  dbms_output.put_line('Salariul mediu este: '||v_salmediu);
  for i in 100..110 loop
    select salariul into v_sal from angajati where id_angajat=i;
    dbms_output.put_line('Salariatul cu codul '||i||' are salariul:
  '||v_sal);
    exit when v_sal<v_salmediu;
  end loop;
end;
```

Structuri LOOP imbricate (se vor eticheta loop-urile)

```
BEGIN
  <<LOOP_EXTERN>>
  LOOP
    v_var:=v_var+1;
    EXIT WHEN v_var>10;
    <<LOOP_INTERN>>
    LOOP
      .....
      EXIT LOOP_EXTERN WHEN conditie1;
      EXIT WHEN conditie2;
      .....
    END LOOP;
    .....
  END LOOP;
END;
```

Probleme propuse spre rezolvare

1) Scrieți un program în PL/SQL pentru a afișa primele n numere cu o diferență de 3 și începând de la 1.

Exemplu:

Pentru n = 10 se vor afișa valorile: 1 4 7 10 13 16 19 22 25 28

2) Scrieți un program în PL/SQL pentru a afișa primele 10 numere din șirul lui Fibonacci.

Precizare: numerele din șirul lui Fibonacci sunt: 0, 1, 1, 2, 3, 5, 8,

3) Scrieți un program în PL/SQL pentru a afișa cel mai mare număr între 3 numere întregi.

4) Scrieți un program în PL/SQL pentru a afișa numărul de zile de Duminică aflate între două date calendaristice date.

Exemplu:

Pentru valorile

start_date = '01-SEP-19'

end_date = '29-SEP-19'

Se obține următorul rezultat:

Numărul de zile de Duminică aflate între cele două date calendaristice este: 5

5) Scrieți un program în PL/SQL pentru a verifica dacă un număr este palindrom sau nu.

Exemplu:

Pentru valoarea n = 12321, se va afișa mesajul ,Numărul este PALINDROM'

Pentru valoarea n = 12322, se va afișa mesajul ,Numărul NU este PALINDROM'

6) Scrieți un program în PL/SQL pentru a afișa fiecare cifră a unui număr în cuvântul corespunzător cifrei.

Exemplu:

Pentru valoarea n = 12345,

se va afișa: Unu Doi Trei Patru Cinci

7) Scrieți un program în PL/SQL pentru a afișa toate valorile între 1000 și 2500 folosind o structură repetitivă WHILE.

8) Scrieți un program în PL/SQL pentru a afișa toate valorile între 1000 și 2500 folosind o structură repetitivă FOR.

Bibliografie web:

<https://www.w3resource.com/>

<https://www.bullraider.com/database/pl-sql/pl-sql-examples>

<https://www.oracletutorial.com/plsql-tutorial/>