

## Laborator 5:

### CURSORI (partea a I-a)

Atunci când se execută o comandă SQL, Oracle Server deschide o zonă de memorie (context area) în care comanda este executată. Cursorul este un pointer către această zonă

În PL/SQL se utilizează două tipuri de cursoare:

1. **implicit**: declarat pentru toate instrucțiunile PL/SQL de tip LMD (INSERT/UPDATE/DELETE);
2. **explicit**: declarat și gestionat de programator.

#### 1. CURSORUL IMPLICIT

Este declarat de PL/SQL implicit pentru toate comenzile de manipulare a datelor (INSERT, UPDATE, DELETE);

Dacă o instrucțiune LMD nu afectează nici o linie a tabelului nu se generează eroare, însă excepția trebuie tratată folosind atributele speciale ale cursorilor;

Atributele cursorului implicit:

- **SQL%ROWCOUNT**
- **SQL%FOUND**
- **SQL%NOTFOUND**

Exemplu

Scrieți un program în PL/SQL pentru a șterge din copia tabelului EMP angajatul cu ID-ul 7788 și a afișa numărul de rânduri șterse, folosind atributul **%ROWCOUNT**.

```
DECLARE
  v_rez NUMBER(2);
BEGIN
  DELETE FROM copy_emp
  WHERE empno=7788;

  v_rez:=SQL%ROWCOUNT;
  DBMS_OUTPUT.PUT_LINE (v_rez || ' randuri sterse');
END;
```

Scriti un program in PL/SQL pentru a arata utilizarea unui cursor implicit fara a folosi niciun atribut.

```
DECLARE
emp_ename VARCHAR2(35);
emp_sal VARCHAR2(35);
zemp_id emp.empno%type:=234;
BEGIN
SELECT ename, sal
INTO emp_ename, emp_sal
FROM emp
WHERE empno = zemp_id;
dbms_output.Put_line ('Employee name: '|| emp_ename|| ' '||emp_sal);
EXCEPTION
WHEN no_data_found THEN
dbms_output.Put_line ('There is no employee with the ID
'||to_char(zemp_id));
END;
```

The screenshot displays the Oracle APEX SQL Workshop interface. At the top, the navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below this, the 'SQL Commands' section is active, showing a dropdown for 'Language' set to 'SQL' and 'Rows' set to '10'. The main area contains the following PL/SQL code:

```
1 DECLARE
2   emp_ename VARCHAR2(35);
3   emp_sal   VARCHAR2(35);
4   zemp_id   emp.empno%type:=234;
5 BEGIN
6   SELECT ename, sal
7   INTO   emp_ename, emp_sal
8   FROM   emp
9   WHERE  empno = zemp_id;
10
11   dbms_output.Put_line ('Employee name: '|| emp_ename|| ' '||emp_sal);
12 EXCEPTION
13   WHEN no_data_found THEN
14     dbms_output.Put_line ('There is no employee with the ID '||to_char(zemp_id));
15 END;
```

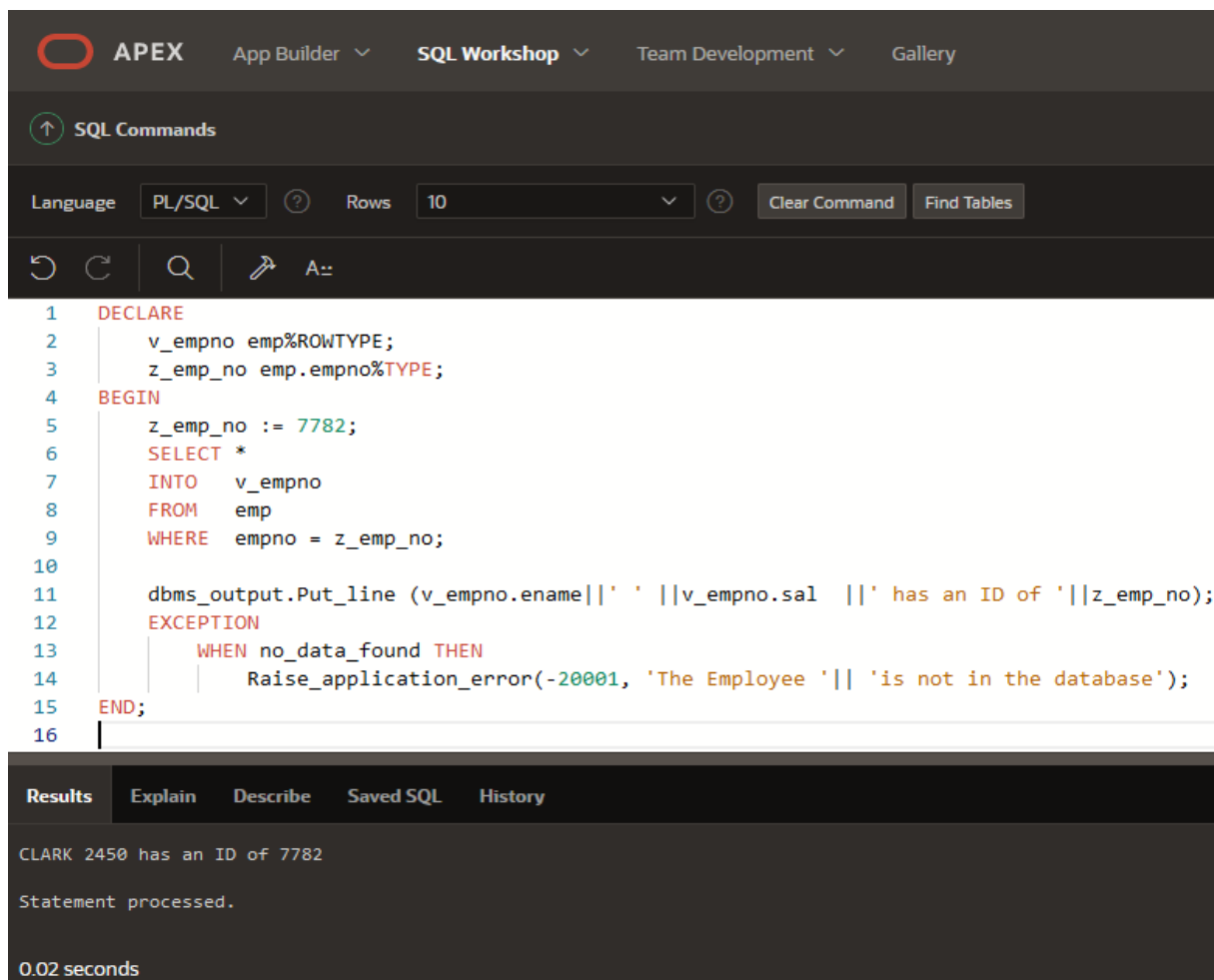
Below the code editor, the 'Results' tab is selected, showing the output of the query:

```
There is no employee with the ID 234
Statement processed.
0.01 seconds
```

Scrieti un program in PL/SQL pentru a crea o inregistrare din tabela cu ajutorul atributului **%ROWTYPE**.

```
DECLARE
    v_empno emp%ROWTYPE;
    z_emp_no emp.empno%TYPE;
BEGIN
    z_emp_no := 7782;
    SELECT *
    INTO v_empno
    FROM emp
    WHERE empno = z_emp_no;

    dbms_output.Put_line (v_empno.ename||' ' ||v_empno.sal ||' has an ID of
'||z_emp_no);
    EXCEPTION
        WHEN no_data_found THEN
            Raise_application_error(-20001, 'The Employee '|| 'is not in the
database');
END;
```



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' window is active, showing the PL/SQL code from the previous block. The code is executed, and the output is displayed in the 'Results' window. The output shows 'CLARK 2450 has an ID of 7782' and 'Statement processed.' The execution time is 0.02 seconds.

```
1 DECLARE
2     v_empno emp%ROWTYPE;
3     z_emp_no emp.empno%TYPE;
4 BEGIN
5     z_emp_no := 7782;
6     SELECT *
7     INTO v_empno
8     FROM emp
9     WHERE empno = z_emp_no;
10
11     dbms_output.Put_line (v_empno.ename||' ' ||v_empno.sal ||' has an ID of '||z_emp_no);
12 EXCEPTION
13     WHEN no_data_found THEN
14         Raise_application_error(-20001, 'The Employee '|| 'is not in the database');
15 END;
```

Results Explain Describe Saved SQL History

CLARK 2450 has an ID of 7782

Statement processed.

0.02 seconds

## 2. CURSORUL EXPLICIT

- ✓ se folosește pentru a procesa individual fiecare linie (înregistrare) returnată de o instrucțiune SELECT ce returnează mai multe înregistrări.
- ✓ mulțimea înregistrărilor returnate de o instrucțiune SELECT este numită mulțime rezultat.
- ✓ cursorul păstrează un pointer către linia curentă în cadrul unei mulțimi rezultat.

Verificarea stării unui cursor explicit se realizează prin intermediul următoarelor atribute:

1. nume\_cursor **%ISOPEN** - evaluat la TRUE în cazul în care cursorul este deschis;
2. nume\_cursor **%NOTFOUND** - evaluat la TRUE în cazul în care cel mai recent FETCH nu a returnat nici o linie;
3. nume\_cursor **%FOUND** - complementul lui %NOTFOUND;
4. nume\_cursor **%ROWCOUNT** - are ca valoare numărul liniilor returnate până în momentul curent.

Prelucrarea cursorului explicit presupune parcurgerea următoarelor etape:

1) se declară variabilele în care vor fi încărcate valorile corespunzătoare unei linii din cursor;

2) se declară cursorul explicit, specificându-se un nume pentru acesta și definindu-se interogarea de procesat în cadrul lui:

**DECLARE nume\_cursor IS SELECT.....;**

3) se deschide cursorul prin intermediul instrucțiunii OPEN, care execută interogarea și legarea tuturor variabilelor referite. Înregistrările returnate de interogare sunt desemnate drept set activ de date, care pot fi de acum încărcate.

**OPEN nume\_cursor;**

4) utilizându-se instrucțiunea FETCH, se încarcă linia curentă din cursor în variabile. Fiecare încărcare determină mutarea pointerului cursorului la linia următoare din setul activ de date.

**FETCH nume\_cursor INTO var1, var2,.....;**

5) este închis cursorul prin instrucțiunea CLOSE, care dezafectează setul activ de linii. Cursorul poate fi din nou deschis pentru a stabili un nou set activ de linii.

**CLOSE nume\_cursor;**

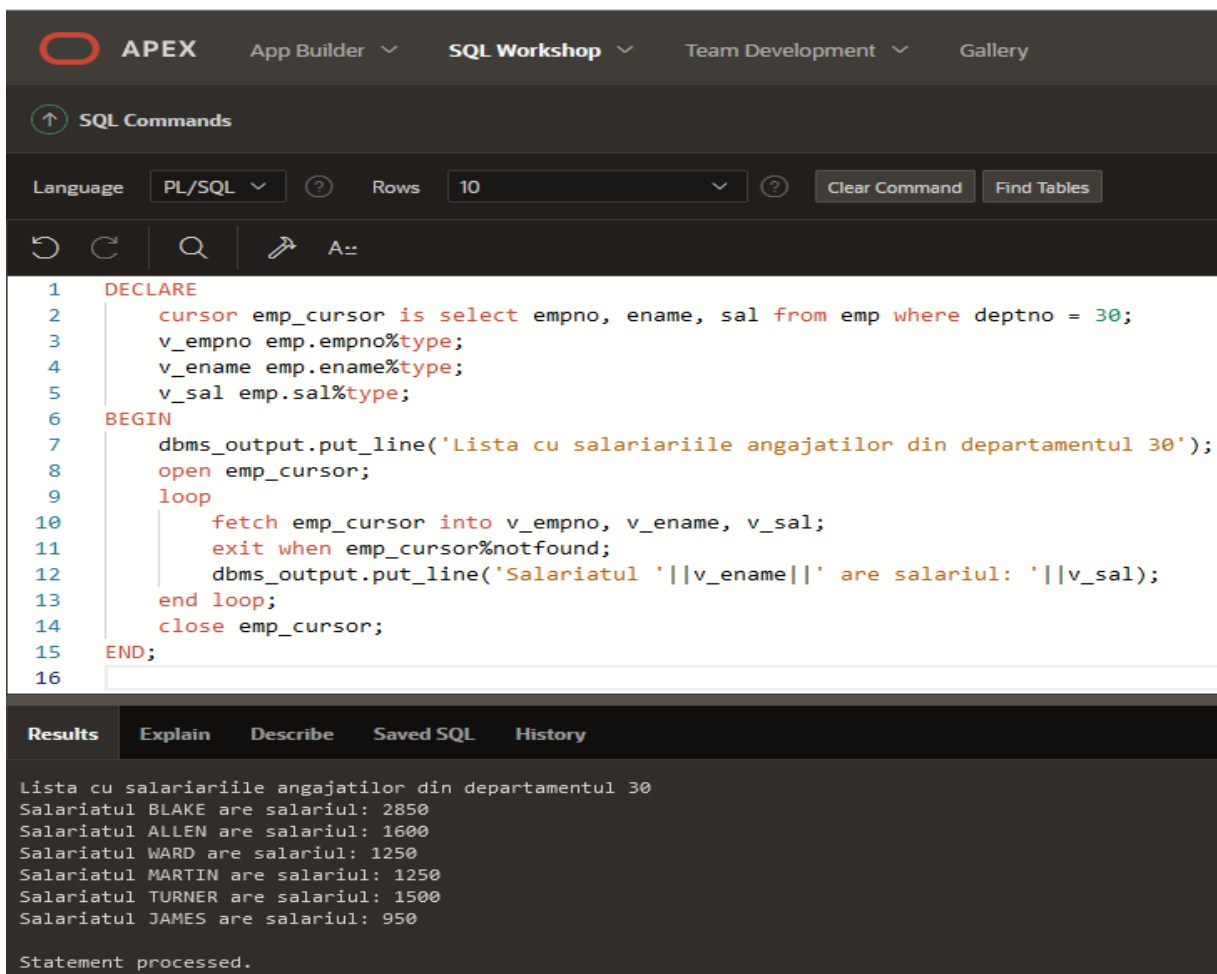
Pentru a procesa liniile unui cursor explicit se definește de obicei o buclă pentru executarea unui FETCH în fiecare iterație. În final, toate liniile din setul activ sunt procesate și un FETCH executat fără succes poziționează atributul %NOTFOUND pe TRUE.

Înainte primului FETCH, %NOTFOUND se evaluează la NULL, ca și în cazul în care FETCH nu se execută niciodată cu succes.

Exemple:

Scrieti un program in PL/SQL pentru a afișa lista cu numele și salariul angajaților din departamentul 30 folosind un *cursor explicit*.

```
DECLARE
    cursor emp_cursor is select empno, ename, sal from emp where deptno = 30;
    v_empno emp.empno%type;
    v_ename emp.ename%type;
    v_sal emp.sal%type;
BEGIN
    dbms_output.put_line('Lista cu salariarile angajatilor din departamentul
30');
    open emp_cursor;
    loop
        fetch emp_cursor into v_empno, v_ename, v_sal;
        exit when emp_cursor%notfound;
        dbms_output.put_line('Salariatul '||v_ename||' are salariul: '||v_sal);
    end loop;
    close emp_cursor;
END;
```



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below the navigation bar, the 'SQL Commands' section is active, showing the PL/SQL code from the example. The code is as follows:

```
1 DECLARE
2     cursor emp_cursor is select empno, ename, sal from emp where deptno = 30;
3     v_empno emp.empno%type;
4     v_ename emp.ename%type;
5     v_sal emp.sal%type;
6 BEGIN
7     dbms_output.put_line('Lista cu salariarile angajatilor din departamentul 30');
8     open emp_cursor;
9     loop
10        fetch emp_cursor into v_empno, v_ename, v_sal;
11        exit when emp_cursor%notfound;
12        dbms_output.put_line('Salariatul '||v_ename||' are salariul: '||v_sal);
13    end loop;
14    close emp_cursor;
15 END;
```

The 'Results' tab is selected, displaying the output of the PL/SQL program:

```
Lista cu salariarile angajatilor din departamentul 30
Salariatul BLAKE are salariul: 2850
Salariatul ALLEN are salariul: 1600
Salariatul WARD are salariul: 1250
Salariatul MARTIN are salariul: 1250
Salariatul TURNER are salariul: 1500
Salariatul JAMES are salariul: 950

Statement processed.
```

Pentru o flexibilitate mai mare se poate utiliza o **variabilă de tip record** pentru încărcarea valorilor din cursor. Această variabilă de tip record poate avea aceleași atribute ca și cursorul prin specificarea proprietății **%ROWTYPE**.

În acest caz încărcarea din cursor se va face direct prin instrucțiunea `fetch var_cursor into var_record`.

Exemplul de mai sus poate fi rescris astfel:

#### DECLARE

```
cursor emp_cursor is select empno, ename, sal from emp where deptno = 30;
--tipul record pt incarcarea valorilor cursorului
emp_rec emp_cursor%rowtype;
```

#### BEGIN

```
dbms_output.put_line('Lista cu salariile angajatilor din departamentul 30');
```

```
open emp_cursor;
```

#### loop

```
fetch emp_cursor into emp_rec;
```

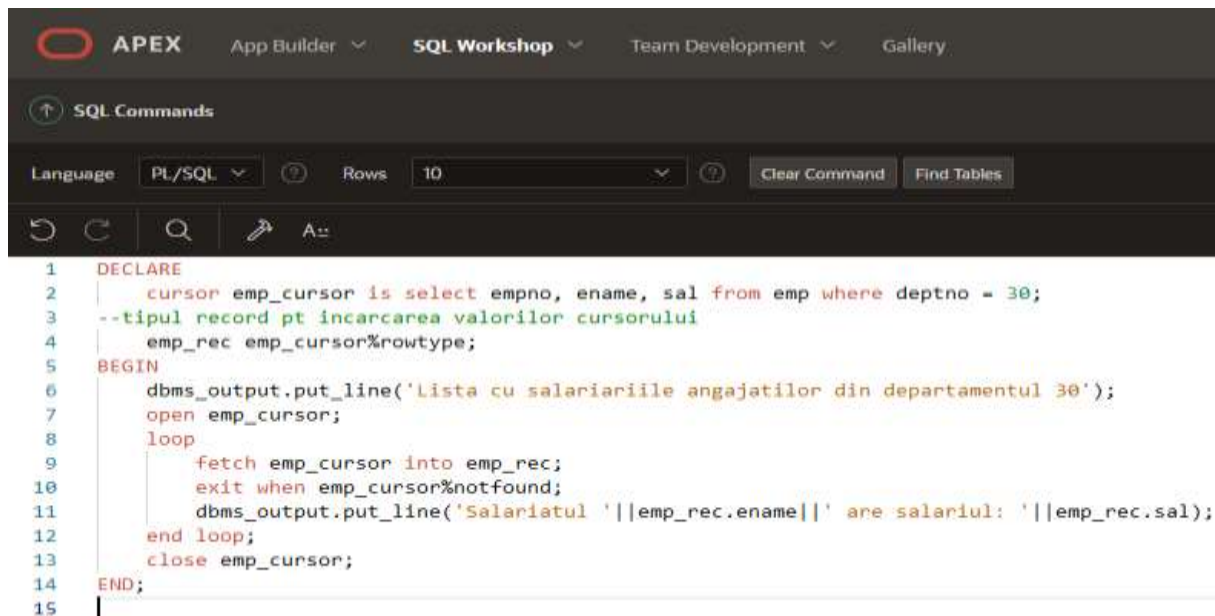
```
exit when emp_cursor%notfound;
```

```
dbms_output.put_line('Salariatul '||emp_rec.ename||' are salariul: '||emp_rec.sal);
```

```
end loop;
```

```
close emp_cursor;
```

```
END;
```



The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below the navigation bar, there is a 'SQL Commands' section with a search icon and a dropdown menu for 'Language' set to 'PL/SQL'. There are also buttons for 'Clear Command' and 'Find Tables'. The main area displays the PL/SQL code from the previous block, with line numbers 1 through 15 on the left. The code is as follows:

```
1 DECLARE
2     cursor emp_cursor is select empno, ename, sal from emp where deptno = 30;
3     --tipul record pt incarcarea valorilor cursorului
4     emp_rec emp_cursor%rowtype;
5 BEGIN
6     dbms_output.put_line('Lista cu salariile angajatilor din departamentul 30');
7     open emp_cursor;
8     loop
9         fetch emp_cursor into emp_rec;
10        exit when emp_cursor%notfound;
11        dbms_output.put_line('Salariatul '||emp_rec.ename||' are salariul: '||emp_rec.sal);
12    end loop;
13    close emp_cursor;
14 END;
```



The screenshot shows the 'Results' section of the Oracle APEX SQL Workshop interface. The top navigation bar includes 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The main area displays the output of the PL/SQL code execution, which is as follows:

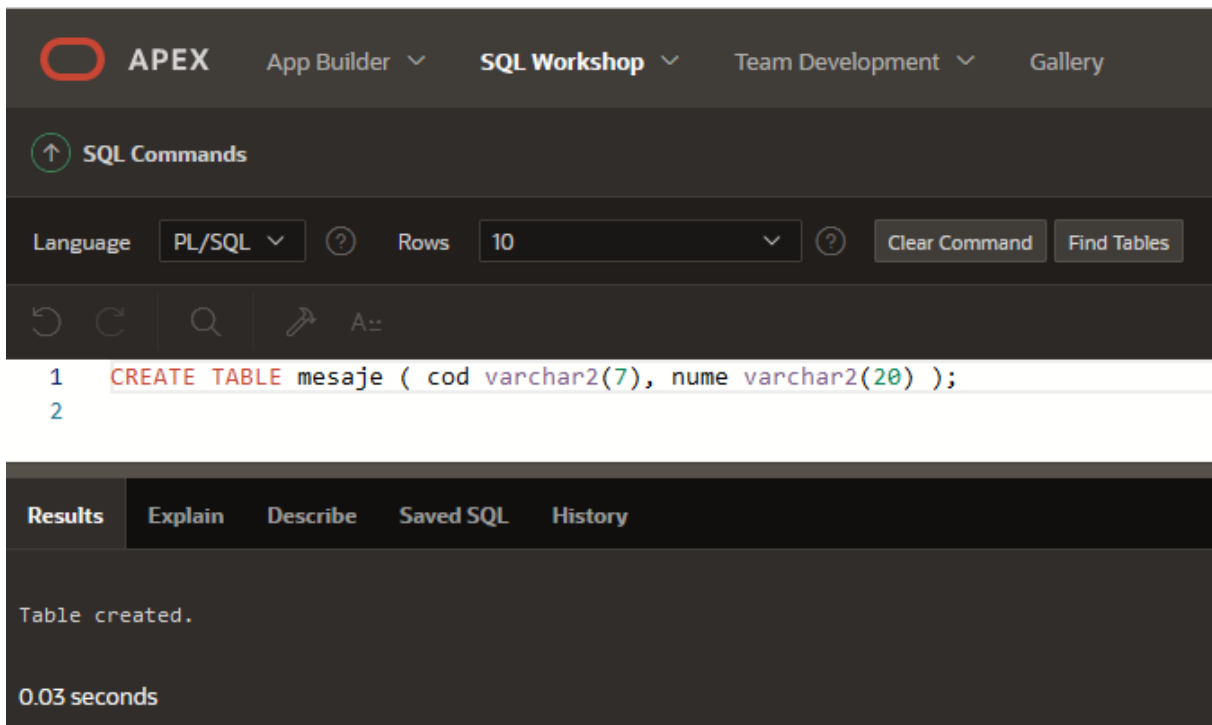
```
Lista cu salariile angajatilor din departamentul 30
Salariatul BLAKE are salariul: 2850
Salariatul ALLEN are salariul: 1600
Salariatul WARD are salariul: 1250
Salariatul MARTIN are salariul: 1250
Salariatul TURNER are salariul: 1500
Salariatul JAMES are salariul: 950
```

Statement processed.

Scrieti un program in PL/SQL pentru a se încarca în tabela MESAJE primii 5 angajați (empno și ename)

Pasul 1 – crearea unei noi tabele numite MESAJE:

```
CREATE TABLE mesaje ( cod varchar2(7), nume varchar2(20) );
```



The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation tabs for 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below this is a 'SQL Commands' section with a search icon and a list of commands. The 'Language' is set to 'PL/SQL' and 'Rows' is set to '10'. There are buttons for 'Clear Command' and 'Find Tables'. The main area displays the SQL command: `1 CREATE TABLE mesaje ( cod varchar2(7), nume varchar2(20) );` and `2`. Below the command, there is a 'Results' section with tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing the message 'Table created.' and the execution time '0.03 seconds'.

Pasul 2 – crearea unui cursor pentru a prelua din tabela EMP primele 5 linii:

```
DECLARE  
    v_id angajati.id_angajat%type;  
    v_nume angajati.nume%type;  
    CURSOR c1 IS SELECT id_angajat, nume FROM angajati;  
BEGIN  
    OPEN c1;  
    FOR i IN 1..5 LOOP  
        FETCH c1 INTO v_id, v_nume;  
        INSERT INTO mesaje VALUES(v_id, v_nume);  
    END LOOP;  
    CLOSE c1;  
END;
```

**SQL Commands**

Language: PL/SQL Rows: 10

```
1 DECLARE
2     v_empno emp.empno%type;
3     v_ename emp.ename%type;
4     CURSOR cursor1 IS SELECT empno, ename FROM emp;
5 BEGIN
6     OPEN cursor1;
7     FOR i IN 1..5 LOOP
8         FETCH cursor1 INTO v_empno, v_ename;
9         INSERT INTO mesaje VALUES(v_empno, v_ename);
10    END LOOP;
11    CLOSE cursor1;
12 END;
```

**Results** Explain Describe Saved SQL History

1 row(s) inserted.

0.05 seconds

Pasul 3 – afisarea continutului noii tabele MESAJE:

**SELECT \* FROM mesaje;**

**SQL Commands**

Language: PL/SQL Rows: 10

```
1 SELECT * FROM mesaje;
2
```

**Results** Explain Describe Saved SQL History

COD	NUME
7839	KING
7698	BLAKE
7782	CLARK
7566	JONES
7388	SCOTT

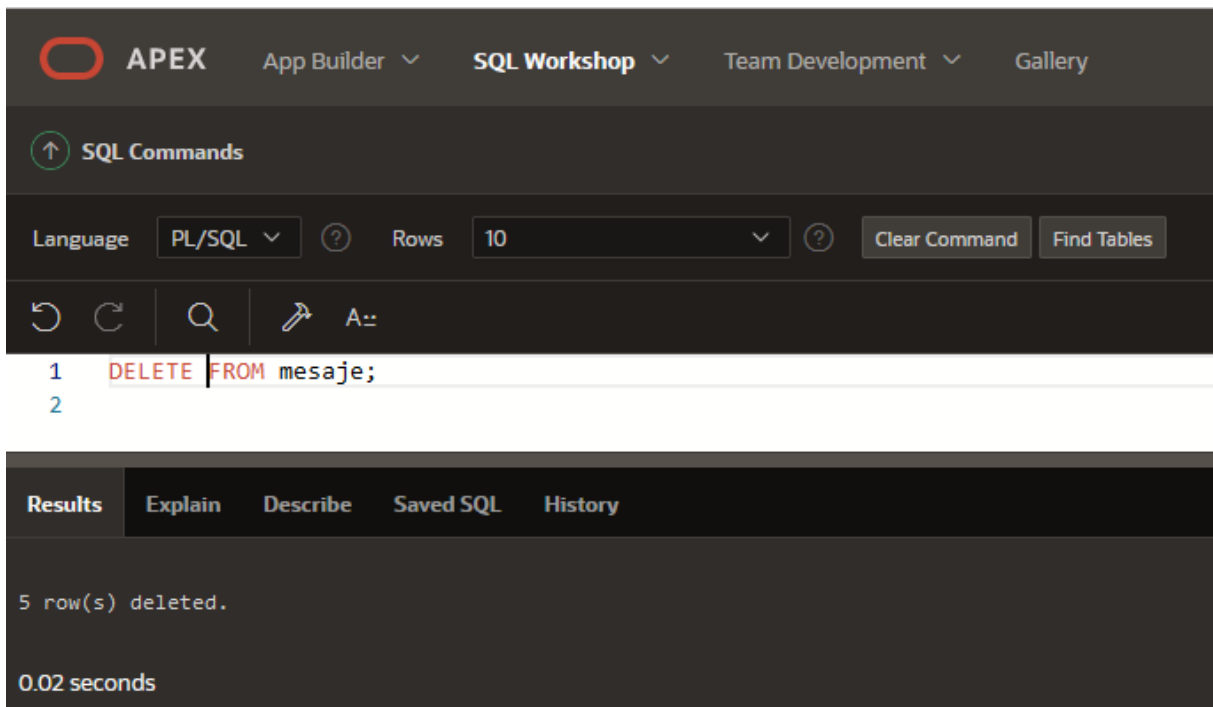
5 rows returned in 0.01 seconds



Testul de ieșire din buclă în acest caz se poate face și cu ajutorul atributului **%ROWCOUNT**:  
Delete from mesaje;

Pasul 1 – Stergerea datelor existente in tabela MESAJE

```
DELETE FROM mesaje;
```



The screenshot shows the Oracle APEX SQL Workshop interface. At the top, there are navigation tabs for 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below this is a 'SQL Commands' section with a search icon and a list of commands. The 'Language' is set to 'PL/SQL' and 'Rows' is set to '10'. There are buttons for 'Clear Command' and 'Find Tables'. Below the command input area, the SQL command `DELETE FROM mesaje;` is entered on line 1. Below the command input area, there are icons for undo, redo, search, and a prompt 'A:'. Below the command input area, the results section is visible, showing '5 row(s) deleted.' and '0.02 seconds'.

Pasul 2 – crearea unui cursor pentru a prelua din tabela EMP primele 5 linii, folosind atributul **%ROWCOUNT**:

```
DECLARE  
  v_empno emp.empno%type;  
  v_ename emp.ename%type;  
CURSOR cursor1 IS SELECT empno, ename FROM emp;  
BEGIN  
  OPEN cursor1;  
  LOOP  
    FETCH cursor1 INTO v_empno, v_ename;  
    EXIT WHEN cursor1%ROWCOUNT>5 OR cursor1%NOTFOUND;  
    INSERT INTO mesaje VALUES (v_empno, v_ename);  
  END LOOP;  
  CLOSE cursor1;  
END;
```

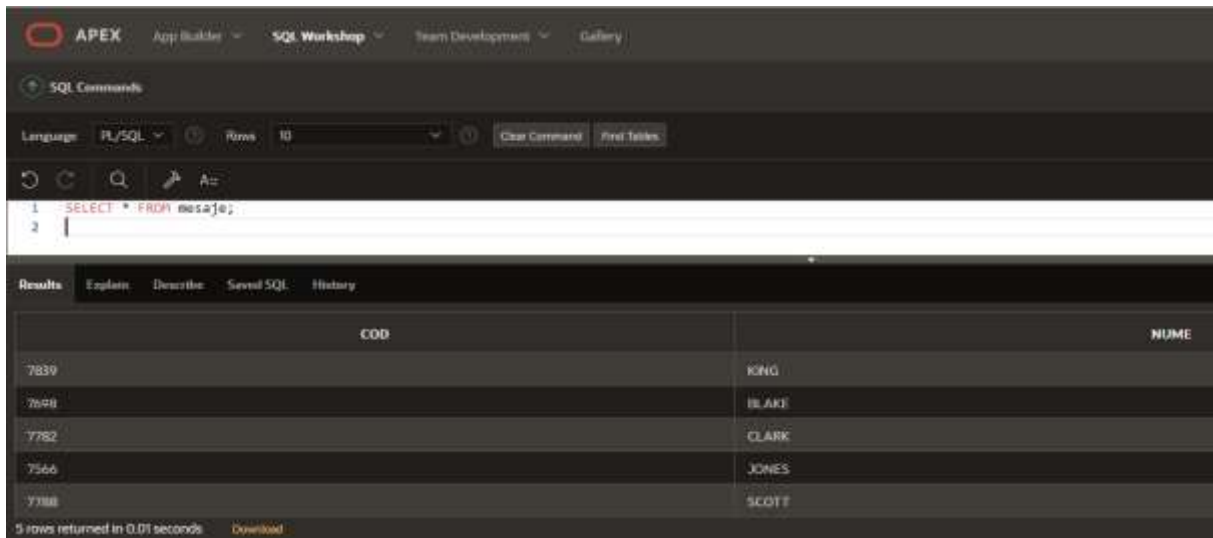
The screenshot displays the Oracle APEX SQL Workshop interface. At the top, the navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below this, the 'SQL Commands' section is active, showing a 'Language' dropdown set to 'PL/SQL' and a 'Rows' limit of '10'. The main area contains a PL/SQL script with the following code:

```
1 DECLARE
2     v_empno emp.empno%type;
3     v_ename emp.ename%type;
4     CURSOR cursor1 IS SELECT empno, ename FROM emp;
5 BEGIN
6     OPEN cursor1;
7     LOOP
8         FETCH cursor1 INTO v_empno, v_ename;
9         EXIT WHEN cursor1%ROWCOUNT>5 OR cursor1%NOTFOUND;
10        INSERT INTO mensaje VALUES (v_empno, v_ename);
11    END LOOP;
12 CLOSE cursor1;
13 END;
```

Below the code editor, the 'Results' tab is selected, showing the output: '1 row(s) inserted.' and a execution time of '0.01 seconds'.

Pasul 3 – afisarea continutului noii tabele MESAJE:

```
SELECT * FROM mensaje;
```



Scrieti un program in PL/SQL pentru a afisa, cu ajutorul unui cursor, toate informatiile despre angajati din tabela EMP.

```

DECLARE
  CURSOR c_emp_info IS
    SELECT empno, ename, sal, deptno
    FROM emp;
  r_emp_info c_emp_info%ROWTYPE;
BEGIN
  OPEN c_emp_info;
  LOOP
    FETCH c_emp_info INTO r_emp_info;
    EXIT WHEN c_emp_info%NOTFOUND;
    dbms_output.Put_line('Employees Information:: ' || ' ID: ' ||r_emp_info.empno
  ||' Name: ' ||r_emp_info.ename || ' ' ||r_emp_info.sal||' ' ||r_emp_info.deptno);
  END LOOP;
  dbms_output.Put_line('Total number of rows : ' ||c_emp_info%rowcount);
  CLOSE c_emp_info;
END;

```



Results	Explain	Describe	Save SQL	History
Employees Information::	ID: 7839	Name: KING	5000	10
Employees Information::	ID: 7698	Name: BLAKE	2850	30
Employees Information::	ID: 7782	Name: CLARK	2450	10
Employees Information::	ID: 7566	Name: JONES	5272.5	20
Employees Information::	ID: 7788	Name: SCOTT	3300	20
Employees Information::	ID: 7902	Name: FORD	3900	20
Employees Information::	ID: 7369	Name: SMITH	800	20
Employees Information::	ID: 7499	Name: ALLEN	1600	30
Employees Information::	ID: 7521	Name: WARD	1250	30
Employees Information::	ID: 7654	Name: MARTIN	1250	30
Employees Information::	ID: 7844	Name: TURNER	1500	30
Employees Information::	ID: 7876	Name: ADAMS	1100	20
Employees Information::	ID: 7900	Name: JAMES	950	30
Employees Information::	ID: 7634	Name: MILLER	1300	10
Total number of rows: 14				
Statement processed.				
0.01 seconds				

## Probleme propuse spre rezolvare

1. Scrieti un program in PL/SQL pentru a declara o inregistrare cu acelasi tip de date ale campurilor din tabela EMP, folosind atributul **%TYPE**.
2. Scrieti un program in PL/SQL pentru a crea un cursor implicit folosind structura repetitiva FOR LOOP.
3. Scrieti un program in PL/SQL pentru a crea un cursor explicit folosind structura repetitiva FOR LOOP.
4. Scrieti un program in PL/SQL pentru a mari salariul angajatilor din departamentul 20 folosind clauza WHERE CURRENT OF intr-un cursor. Marirea salariilor se face astfel: Daca salariul este mai mic de 7000 se va mari cu 15%, altfel cu se va mari cu 10%.
5. Scrieti un program in PL/SQL pentru a prelua doar o singura linie din tabela EMP, cu ajutorul instructiunii FETCH.
6. Scrieti un program in PL/SQL pentru a prelua mai mult de o singura linie din tabela EMP, cu ajutorul instructiunii FETCH.
7. Scrieti un program in PL/SQL pentru a afisa numele si salariul fiecarui angajat din tabela EMP al carui salariu este mai mic decat o valoare transmisa ca parametru intr-un cursor.
8. Scrieti un program in PL/SQL pentru a afisa denumirea departamentului, seful departamentului, data angajarii si salariul fiecarui angajat din tabela EMP al carui salariu este cel mai mare.
9. Scrieti un program in PL/SQL pentru a arata cum se utilizeaza un subquery (subcerere) in clauza FROM din interogarea principala a unui cursor explicit.

Bibliografie web:

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

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

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