

Laborator 6: CURSORI (partea a II-a)

1. CURSORUL EXPLICIT

Gestiunea implicită a cursorului prin utilizarea unui ciclu FOR:

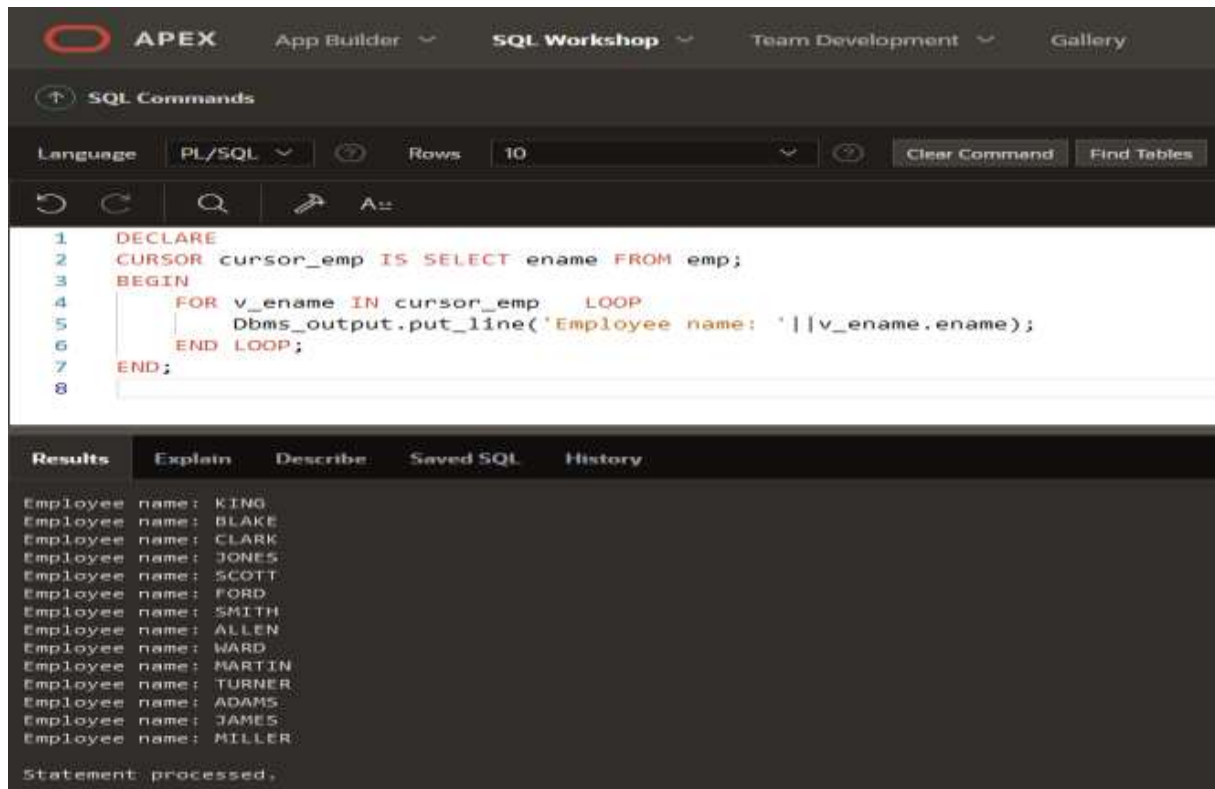
FOR nume_record **IN** nume_cursor **LOOP**

....
END LOOP;

În acest caz, tipul RECORD nu trebuie declarat. Se realizează în mod implicit deschiderea, încărcarea și închiderea cursorului.

Exemplu - Scrieti un program in PL/SQL pentru a afișa printr-un ciclu FOR numele angajaților din tabela EMP:

```
DECLARE
CURSOR cursor_emp IS SELECT ename FROM emp;
BEGIN
  FOR v_ename IN cursor_emp LOOP
    Dbms_output.put_line('Employee name: '||v_ename.ename);
  END LOOP;
END;
```



The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The main area shows the 'SQL Commands' editor with the following PL/SQL code:

```
1 DECLARE
2   CURSOR cursor_emp IS SELECT ename FROM emp;
3 BEGIN
4   FOR v_ename IN cursor_emp LOOP
5     Dbms_output.put_line('Employee name: '||v_ename.ename);
6   END LOOP;
7 END;
```

The 'Results' tab is active, showing the output of the program:

```
Employee name: KING
Employee name: BLAKE
Employee name: CLARK
Employee name: JONES
Employee name: SCOTT
Employee name: FORD
Employee name: SMITH
Employee name: ALLEN
Employee name: WARD
Employee name: MARTIN
Employee name: TURNER
Employee name: ADAMS
Employee name: JAMES
Employee name: MILLER

Statement processed.
```

Utilizarea unui cursor direct în cadrul instrucțiunii FOR.

În acest caz cursorul nu este declarat, nu are nume, este reprezentat doar de interogarea SELECT din cadrul instrucțiunii FOR, astfel:

FOR NUME_RECORD IN (SELECT.....) LOOP

....
END LOOP;

Dezavantajul în acest caz este ca nu se pot utiliza atributele cursorului din cauza faptului că acesta nu are nume.

Exemplu - Scrieti un program in PL/SQL pentru a afișa suma aferentă salariilor din fiecare departament:

```
BEGIN
  dbms_output.put_line('Total salarii pe fiecare departament:');
  for dep_rec in
    ( select d.deptno dep, sum(a.sal) sal
      from emp a, dept d
      where a.deptno=d.deptno
      group by d.deptno)
  loop
    dbms_output.put_line('Departamentul '||dep_rec.dep||' are de platit
salarii in valoare de: '||dep_rec.sal||' RON');
  end loop;
END;
```

```
APEX App Builder SQL Workshop Team Development Gallery
SQL Commands
Language PL/SQL Rows 10 Clear Command Find Tables
1 BEGIN
2   dbms_output.put_line('Total salarii pe fiecare departament:');
3   for dep_rec in
4     ( select d.deptno dep, sum(a.sal) sal
5       from emp a, dept d
6       where a.deptno=d.deptno
7       group by d.deptno)
8   loop
9     dbms_output.put_line('Departamentul '||dep_rec.dep||' are de platit salarii in valoare de: '||dep_rec.sal||' RON');
10  end loop;
11 END;
12

Results Explain Describe Saved SQL History
Total salarii pe fiecare departament:
Departamentul 10 are de platit salarii in valoare de: 8750 RON
Departamentul 20 are de platit salarii in valoare de: 12372.5 RON
Departamentul 30 are de platit salarii in valoare de: 9400 RON
Statement processed.
```

Utilizarea cursorilor cu parametri

Pentru o flexibilitate mai mare se pot declara și utiliza cursori cu parametri care transmit valorile parametrilor actuali în cererile SQL. Declararea cursorului cu parametri se face astfel:

Cursor nume_cursor (parametru1 tip_data,.....)

Is select;

Deschidere:

Open nume_cursor(valoare_parametru1,.....);

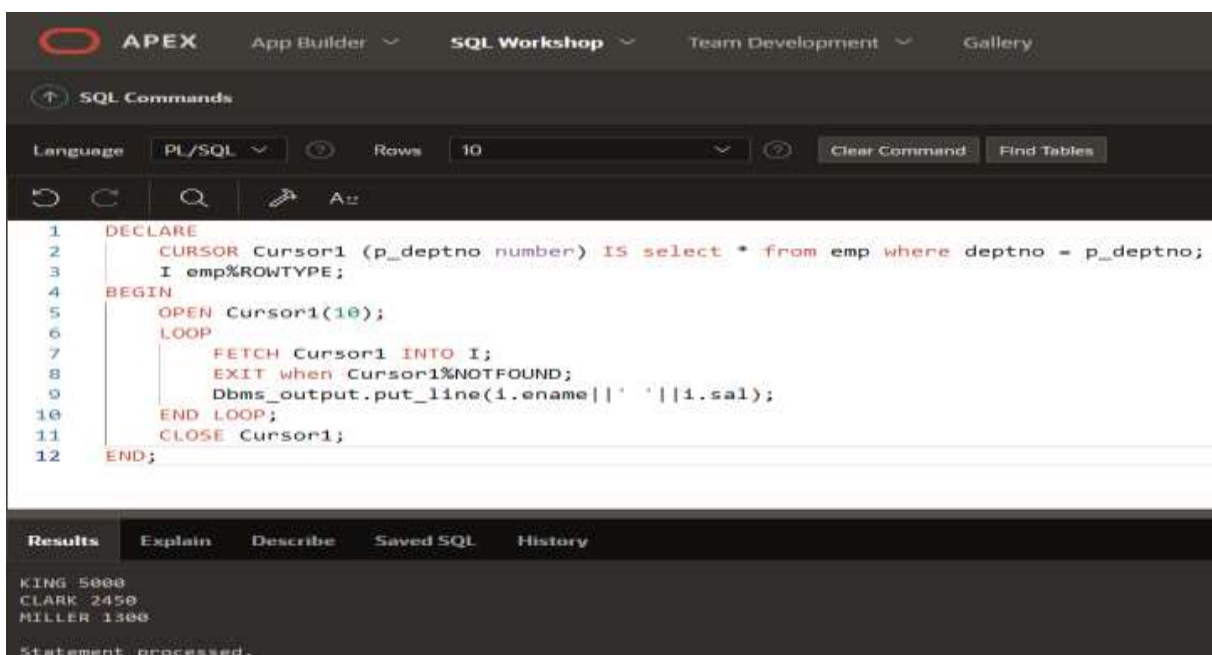
Cursoarele parametrizate nu oferă o funcționalitate suplimentară ci doar o modalitate simplă și clară de a specifica valori de intrare.

Tipurile parametrilor sunt scalare, dar nu li se precizează dimensiunea; ele fiind referite în interogare.

Exemple:

Scrieti un program in PL/SQL pentru a totii angajatii din departamentul 10 cu ajutorul unui cursor cu parametru – departamentul numarul 10.

```
DECLARE
  CURSOR Cursor1 (p_deptno number) IS select * from emp where deptno =
  p_deptno;
  I emp%ROWTYPE;
BEGIN
  OPEN Cursor1(10);
  LOOP
    FETCH Cursor1 INTO I;
    EXIT when Cursor1%NOTFOUND;
    Dbms_output.put_line(i.ename||' '||i.sal);
  END LOOP;
  CLOSE Cursor1;
END;
```



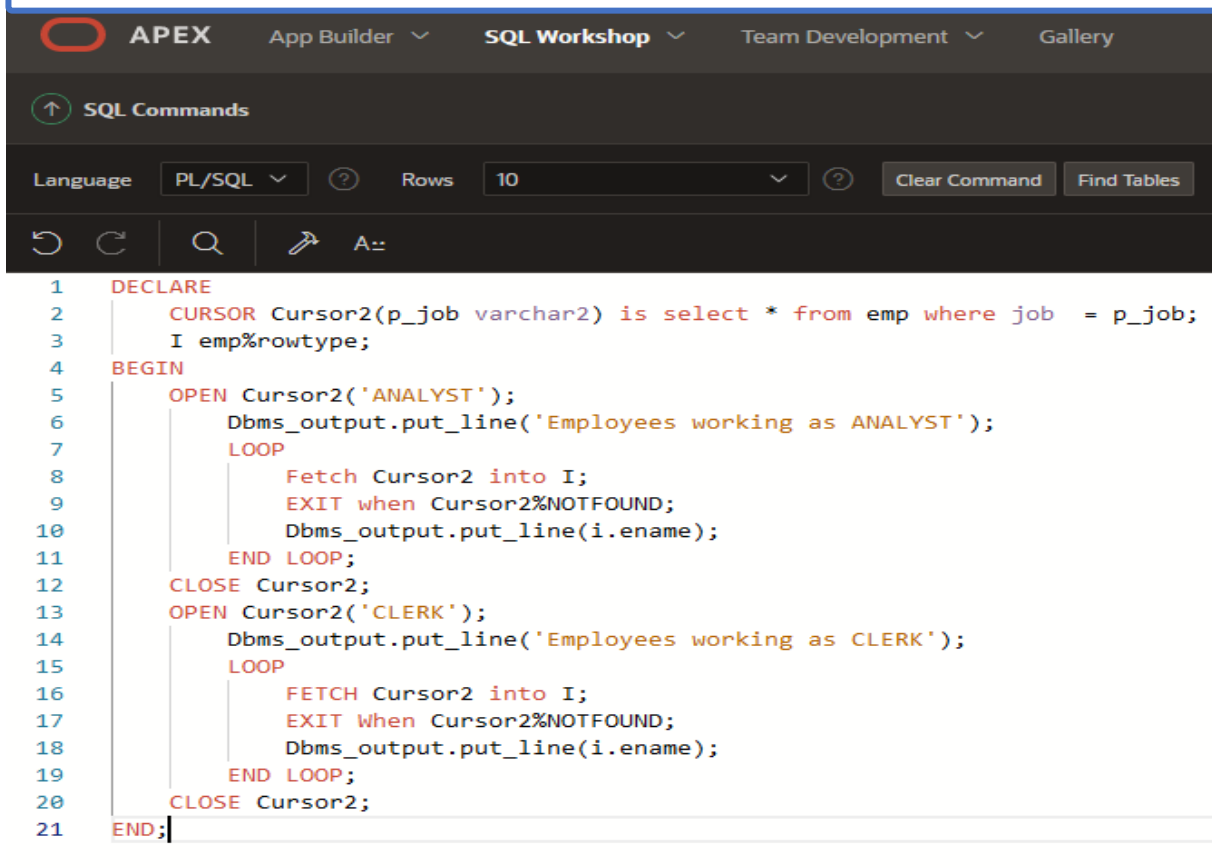
```
1 DECLARE
2   CURSOR Cursor1 (p_deptno number) IS select * from emp where deptno =
3   p_deptno;
4   I emp%ROWTYPE;
5 BEGIN
6   OPEN Cursor1(10);
7   LOOP
8     FETCH Cursor1 INTO I;
9     EXIT when Cursor1%NOTFOUND;
10    Dbms_output.put_line(i.ename||' '||i.sal);
11  END LOOP;
12  CLOSE Cursor1;
13 END;
```

Results

```
KING 5000
CLARK 2450
MILLER 1300
Statement processed.
```

Scrieti un program in PL/SQL pentru a afisa datele despre angajatii care lucreaza ca **ANALYST** sau ca **CLERK**, denumirea job-ului fiind transmisa prin intermediul unui parametru al cursorului.

```
DECLARE
  CURSOR Cursor2(p_job varchar2) is select * from emp where job = p_job;
  I emp%rowtype;
BEGIN
  OPEN Cursor2('ANALYST');
  Dbms_output.put_line('Employees working as ANALYST');
  LOOP
    Fetch Cursor2 into I;
    EXIT when Cursor2%NOTFOUND;
    Dbms_output.put_line(i.ename);
  END LOOP;
  CLOSE Cursor2;
  OPEN Cursor2('CLERK');
  Dbms_output.put_line('Employees working as CLERK');
  LOOP
    FETCH Cursor2 into I;
    EXIT When Cursor2%NOTFOUND;
    Dbms_output.put_line(i.ename);
  END LOOP;
  CLOSE Cursor2;
END;
```



```
1 DECLARE
2   CURSOR Cursor2(p_job varchar2) is select * from emp where job = p_job;
3   I emp%rowtype;
4 BEGIN
5   OPEN Cursor2('ANALYST');
6   Dbms_output.put_line('Employees working as ANALYST');
7   LOOP
8     Fetch Cursor2 into I;
9     EXIT when Cursor2%NOTFOUND;
10    Dbms_output.put_line(i.ename);
11  END LOOP;
12  CLOSE Cursor2;
13  OPEN Cursor2('CLERK');
14  Dbms_output.put_line('Employees working as CLERK');
15  LOOP
16    FETCH Cursor2 into I;
17    EXIT When Cursor2%NOTFOUND;
18    Dbms_output.put_line(i.ename);
19  END LOOP;
20  CLOSE Cursor2;
21 END;
```

```
Results Explain Describe Saved SQL History
Employees working as ANALYST
SCOTT
FORD
Employees working as CLERK
SMITH
ADAMS
JAMES
MILLER

Statement processed.
```

Actualizarea înregistrărilor returnate de cererea cursorului. Clauza FOR UPDATE

- se blochează setul de înregistrări ale cursorului în 2 variante: NOWAIT și WAIT n:

CURSOR C IS SELECT FROM.... FOR UPDATE [OF COLUMN_NAME] [NOWAIT|WAIT n];

- se adaugă clauza FOR UPDATE în interogarea asociată cursorului pentru a bloca liniile afectate atunci când cursorul este deschis.
- clauza NOWAIT - determină apariția unei erori dacă liniile sunt blocate de o altă sesiune.
- când mai multe tabele sunt implicate în interogare, se poate folosi FOR UPDATE pentru a impune blocarea liniilor unei tabele anume. Liniile unei tabele sunt blocate numai în cazul în care clauza FOR UPDATE face o referire la o coloană din acea tabelă.

Exemplu:

Scrieti un program in PL/SQL pentru a afisa salariul marit cu 1000 pentru toti angajatii din tabela EMP.

Pasul 1 – se creeaza un cursor care rezolva cerinta specificata

```
DECLARE
CURSOR Cursor3
IS
  SELECT * FROM EMP FOR UPDATE OF sal nowait;
BEGIN
  FOR Cursor_rec IN Cursor3
  LOOP
    UPDATE EMP SET SAL = SAL+1000 WHERE CURRENT OF Cursor3;
  END LOOP;
END;
```

```
APEX App Builder SQL Workshop Team Development Gallery
SQL Commands
Language PL/SQL Rows 10 Clear Command Find Tables
1 DECLARE
2   CURSOR Cursor3
3   IS
4     SELECT * FROM EMP FOR UPDATE OF sal nowait;
5 BEGIN
6   FOR Cursor_rec IN Cursor3
7   LOOP
8     UPDATE EMP SET SAL = SAL+1000 WHERE CURRENT OF Cursor3;
9   END LOOP;
10 END;
```

Results Explain Describe Saved SQL History
1 row(s) updated.

Pasul 2 – se afiseaza continutul tabelii EMP pentru a vedea modificarile in campul SAL.

SELECT * FROM emp;

The screenshot shows the APEX SQL Workshop interface with the query `SELECT * FROM emp;` entered in the command area. The results are displayed in a table with the following columns: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTNO. The table contains 14 rows of data.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17/09/1981	5000	-	10
7812	BLAKE	MANAGER	7839	01/01/1981	3800	-	10
7844	CLERK	MANAGER	7839	09/09/1981	1400	-	10
7866	JONES	MANAGER	7839	01/07/1981	2975	-	10
7869	SCOTT	ANALYST	7566	07/06/1982	4000	-	10
7876	FORD	ANALYST	7566	03/12/1981	3900	-	10
7899	SMITH	CLERK	7902	07/12/1980	800	-	10
7902	ALLEN	SALESMAN	7909	01/20/1981	1600	300	30
7913	WARD	SALESMAN	7909	01/22/1981	1250	500	30
7924	MARTIN	SALESMAN	7909	01/28/1981	1200	1400	30

Pentru manipularea cât mai ușoară a comenzilor LMD UPDATE și DELETE se poate utiliza clauza WHERE CURRENT OF care permite actualizarea înregistrărilor pe baza liniei curente din cursor.

UPDATE tabela

SET camp=....

WHERE CURRENT OF nume_cursor;

- se poate referi linia din tabela originală, pentru actualizare sau ștergere, prin intermediul liniei curente a cursorului (cea procesată de ultima instrucțiune FETCH).
- clauza FOR UPDATE trebuie inclusă în definiția cursorului pentru a bloca liniile în momentul execuției instrucțiunii OPEN.

Exemplu:

Exemplu pentru actualizarea înregistrărilor din tabela EMP realizându-se cu clauza WHERE CURRENT OF:

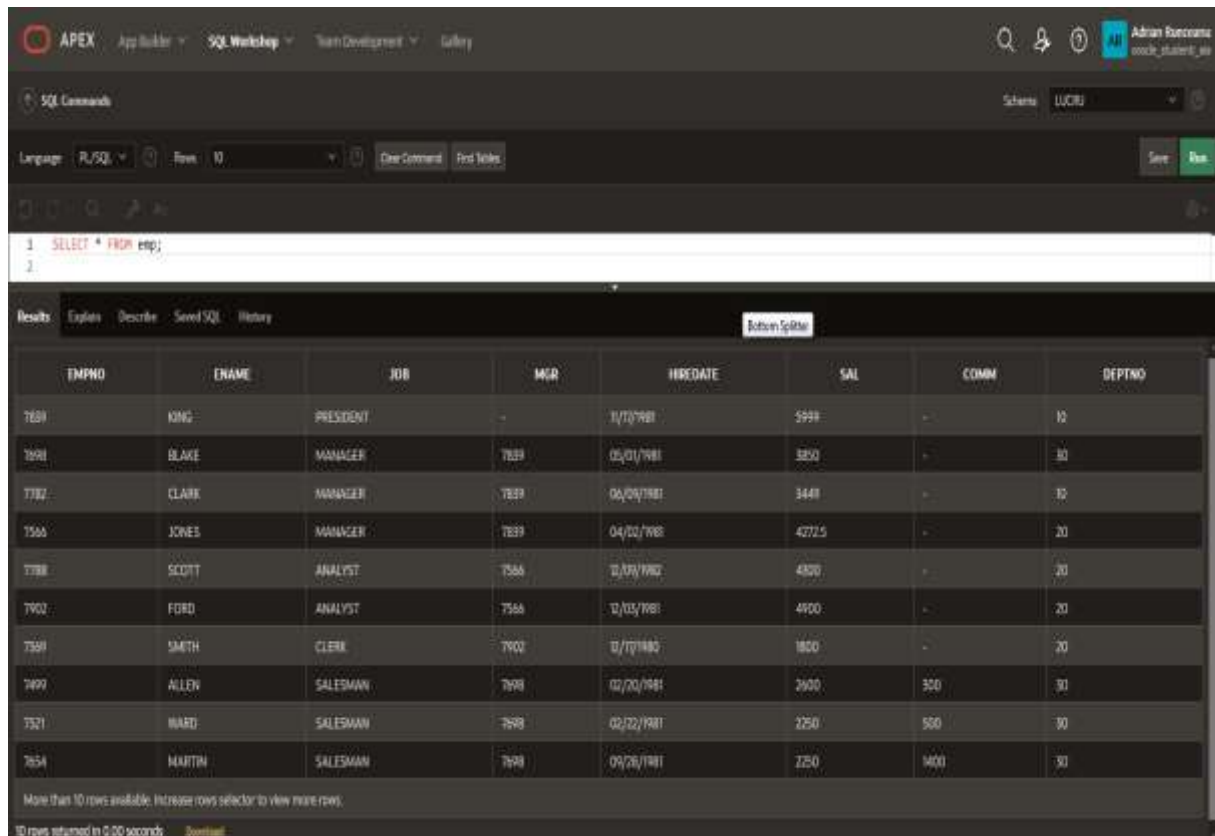
```
DECLARE
  CURSOR emp_cur IS
  SELECT e.ename, d.dname FROM emp e, dept d
  WHERE d.deptno = 10 and e.deptno = d.deptno
  FOR UPDATE of e.sal;
BEGIN
  FOR employee_rec IN emp_cur
  LOOP
    UPDATE emp
      SET sal = sal - 1
      WHERE CURRENT OF emp_cur;
  END LOOP;
END;
```

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 these, the 'SQL Commands' window is active, displaying the PL/SQL code from the previous block. The code is as follows:

```
1 DECLARE
2   CURSOR emp_cur IS
3     SELECT e.ename, d.dname FROM emp e, dept d
4     WHERE d.deptno = 10 and e.deptno = d.deptno
5     FOR UPDATE of e.sal;
6 BEGIN
7   FOR employee_rec IN emp_cur
8   LOOP
9     UPDATE emp
10      SET sal = sal - 1
11      WHERE CURRENT OF emp_cur;
12   END LOOP;
13 END;
```

At the bottom of the interface, the 'Results' window shows the output: '1 row(s) updated.'

```
SELECT * FROM emp;
```



The screenshot shows the Oracle APEX SQL Workshop interface. The SQL command 'SELECT * FROM emp;' is entered in the command window. The results window displays a table with the following columns: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTNO. The table contains 14 rows of employee data.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		11/7/1981	5000	-	10
7812	BLAKE	MANAGER	7839	05/01/1981	3800	-	30
7782	CLARK	MANAGER	7839	09/08/1981	3400	-	10
7566	JONES	MANAGER	7839	04/02/1980	4700	-	20
7788	SCOTT	ANALYST	7566	12/17/1982	4300	-	20
7902	FORD	ANALYST	7566	12/13/1981	4900	-	20
7691	SMITH	CLERK	7902	12/17/1980	1800	-	20
7699	ALLEN	SALESMAN	7698	02/20/1981	2600	350	30
7521	WARD	SALESMAN	7698	02/22/1981	2500	550	30
7654	MARTIN	SALESMAN	7698	09/28/1981	2200	1400	30

Scriti 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;

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language PL/SQL Rows 10 Clear Command Find Tables

```

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 salariile 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;
16

```

Results Explain Describe Saved SQL History

```

Lista cu salariile angajatilor din departamentul 30
Salariatul BLAKE are salariul: 3850
Salariatul ALLEN are salariul: 2600
Salariatul WARD are salariul: 2250
Salariatul MARTIN are salariul: 2250
Salariatul TURNER are salariul: 2500
Salariatul JAMES are salariul: 1950

Statement processed.

```

Scrieti un program in PL/SQL pentru a dubla valoarea salariilor celor angajați înainte de 1 ianuarie 1995.

```

DECLARE
CURSOR before95 IS
SELECT *
FROM copy_emp
WHERE hiredate <= TO_DATE('01-JAN-1995','DD-MON-YYYY')
FOR UPDATE OF sal NOWAIT;
BEGIN
FOR x IN before95 LOOP
UPDATE copy_emp
SET sal = sal * 2
WHERE CURRENT OF before95;
END LOOP;
END;

```



SQL Commands

Language

PL/SQL ▾



Rows

10 ▾



Clear Command

Find Tables



A::

```
1 DECLARE
2     CURSOR before95 IS
3         SELECT *
4         FROM   copy_emp
5         WHERE  hiredate <= TO_DATE('01-JAN-1995','DD-MON-YYYY')
6         FOR UPDATE OF sal NOWAIT;
7 BEGIN
8     FOR x IN before95 LOOP
9         UPDATE copy_emp
10        SET    sal = sal * 2
11        WHERE CURRENT OF before95;
12    END LOOP;
13 END;
14
```

Results

Explain

Describe

Saved SQL

History

1 row(s) updated.

Probleme propuse spre rezolvare

1. Scrieți un program în PL/SQL pentru a prelua primele trei rânduri dintr-o multime de rezultate în trei înregistrări utilizând același cursor explicit în variabile diferite.

Cerinta: Se vor afisa datele in urmatorul format:

```
BLAKE          30 (SALES)
CLARK          10 (ACCOUNTING)
JONES          20 (RESEARCH)
```

2. Scrieți un program în PL/SQL pentru a prelua mai multe înregistrări cu ajutorul unui cursor imbricat.

Cerinta: Se vor afisa datele in urmatorul format:

Departament : ACCOUNTING

Angajat: KING Salariu: 5999
Angajat: CLARK Salariu: 3449
Angajat: MILLER Salariu: 2299

Departament : RESEARCH

Angajat: JONES Salariu: 9674.14
Angajat: SCOTT Salariu: 9734.97
Angajat: FORD Salariu: 9734.97
Angajat: SMITH Salariu: 4788.05
Angajat: ADAMS Salariu: 5586.06

Departament : SALES

Angajat: BLAKE Salariu: 3850
Angajat: ALLEN Salariu: 2600
Angajat: WARD Salariu: 2250
Angajat: MARTIN Salariu: 2250
Angajat: TURNER Salariu: 2500
Angajat: JAMES Salariu: 1950

Departament : OPERATIONS

3. Scrieți program în PL/SQL pentru a afișa numele departamentului și al celui mai costisitor angajat din departament (cu salariul cel mai mare). Afișează numele departamentului și al celui mai scump angajat.

Cerinta: Se vor afisa datele in urmatorul format:

```
ACCOUNTING      KING
RESEARCH        - mai multi angajati
SALES           BLAKE
```

Bibliografie web:

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

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

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