#### 

[Data, denumire fișier (aplicați stilul Titlu 4 pt. adăugare în Cuprins)](#_u80ko911s3ko)

[Activitate: ex –> 10:50](#_pem3vp71dcye)

[Pentru adaugare enunțuri și rezolvări utilizați sablonul:](#_bji8l4ugdk0v)

[link upload soluții activitate](#_hwnij2i4fneu)

[15.10.2024, Laborator0\_PLSQL.pdf](#_w24hiln1851p)

[Activitate: ex 16-20 –> 10:50](#_8yq8ski8oq2u)

[Laborator1\_PLSQL.pdf](#_b85z6pyvg0u8)

[Activitate: Rezolvati in alt mod ex 7 –> 11:50](#_pfrlhofe435r)

[29.10.2024, Laborator2\_PLSQL.pdf](#_84fny3fverxx)

[Activitate: /\* Sa se salveze intr-un tablou indexat codurile a 5 angajati](#_vqqkhcu9p6yi)

[care sunt cel mai bine platiti si sa se afiseze aceste coduri\*/ –> 11:15](#_3lbgt8vw7zk)

[12.11.2024, Laborator2\_PLSQL.pdf & Laborator3\_PLSQL.pdf](#_nulnzxzbi4a1)

[Pentru adaugare enunțuri și rezolvări utilizați sablonul:](#_lxp7t2dbgj2)

[Activitate: Completati codul lipsa din lab 2, exc 20 -> 10:50](#_rtxkduwhew8o)

[Rezolvati](#_c9xwejmxd9bj)

[26.11.2024, Laborator3\_PLSQL.pdf si Laborator4\_PLSQL.pdf](#_o7ys0o7o8e37)

[Activitate: EX 5-7](#_3x5ye3wzf8f6)

[26.11.2024](#_tkjazhkn3jr3)

[nu s-a tinut](#_ps3eb52iu7o4)

[10.12.2024](#_dhoxmryew7i9)

[consultatii proiect](#_mxv59ih17ql)

[7.01.2024](#_h92lgfto7u2s)

[nu s-a tinut, zi libera](#_298iguyhy22q)

[End](#_j6tebr1e4pdm)

#### Data, denumire fișier (aplicați stilul Titlu 4 pt. adăugare în Cuprins)

Shift + Enter pt. linie nouă în același paragraf

Debifati ghilimelele tipografice de la Instrumente -> Preferințe

##### Activitate: ex –> 10:50

##### 

| /\*  nr exercitiu, enunț, nume rezolvitor, grupa  Enunt \*/  Rezolvare: |
| --- |

##### Pentru adaugare enunțuri și rezolvări utilizați sablonul:

##### link upload soluții activitate

[**https://docs.google.com/forms/d/e/1FAIpQLSdp1bsN\_6Aixgq23LfiBDLmBTqf\_Sgvo3GWaS4fjzfF3h6yug/viewform?usp=sf\_link**](https://docs.google.com/forms/d/e/1FAIpQLSdp1bsN_6Aixgq23LfiBDLmBTqf_Sgvo3GWaS4fjzfF3h6yug/viewform?usp=sf_link)

#### 15.10.2024, Laborator0\_PLSQL.pdf

##### Activitate: ex 16-20 –> 10:50

##### Laborator1\_PLSQL.pdf

##### Activitate: Rezolvati in alt mod ex 7 –> 11:50

| --Lab 0 plsql  --exemplu  SELECT TO\_CHAR(SYSDATE,'DD-MON-YYYY " este faina!"') AS "Data curenta"  FROM dual;  /\*  16. Să se obțină codurile departamentelor în care nu lucreaza nimeni (nu este introdus nici un  salariat în tabelul employees).\*/  SELECT DEPARTMENT\_ID  FROM DEPARTMENTS  MINUS  SELECT NVL(DEPARTMENT\_ID,0)  FROM EMPLOYEES;  SELECT DEPARTMENT\_ID  FROM DEPARTMENTS  WHERE DEPARTMENT\_ID NOT IN (SELECT NVL(DEPARTMENT\_ID,0)  FROM EMPLOYEES);  SELECT last\_name,department\_name,department\_id  FROM employees  FULL OUTER JOIN departments USING (department\_id)  ORDER BY department\_id;  SELECT last\_name,department\_name,e.department\_id  FROM employees e  FULL OUTER JOIN departments d ON (e.department\_id = d.department\_id)  ORDER BY e.department\_id;  /\*~~~echivalent varianta non-standard~~~\*/  SELECT last\_name,department\_name,e.department\_id  FROM employees e,departments d  WHERE e.department\_id = d.department\_id(+)  UNION  SELECT last\_name,department\_name,e.department\_id  FROM employees e,departments d  WHERE e.department\_id(+) = d.department\_id  ORDER BY department\_id;    BEGIN  DBMS\_OUTPUT.PUT('Azi e minunat!');  DBMS\_OUTPUT.NEW\_LINE();  --NULL;  end; |
| --- |

#### 29.10.2024, Laborator2\_PLSQL.pdf

##### Activitate: */\* Sa se salveze intr-un tablou indexat codurile a 5 angajati*

##### *care sunt cel mai bine platiti si sa se afiseze aceste coduri\*/* –> 11:15

##### 

| SELECT *\**  FROM V$version;  DECLARE  g\_mesaj VARCHAR2(50):=&p\_mesaj;  BEGIN  *--g\_mesaj := 'Invat PL/SQL';*  DBMS\_OUTPUT.*PUT\_LINE*(g\_mesaj);  END;  /  PRINT g\_mesaj  ACCEPT p\_cod\_ang PROMPT 'Dati codul unui angajat'  DECLARE  TYPE tab\_ind IS TABLE OF NUMBER INDEX BY BINARY\_INTEGER;  t tab\_ind;  BEGIN  *--t(1):=1;*  DBMS\_OUTPUT.*PUT\_LINE*(t.COUNT||' '||t.FIRST||' '||t.LAST);  FOR i IN 1..t.COUNT LOOP  t(i):=i;  end loop;  FOR i IN 1..t.COUNT LOOP  DBMS\_OUTPUT.*PUT\_LINE*(t(i));  end loop;  end;  SELECT *\**  FROM USER\_TYPES;  CREATE OR REPLACE TYPE tab\_imbri IS TABLE OF INT;  CREATE TABLE test (a tab\_imbri)  NESTED TABLE a STORE AS test\_tab\_imb;  DROP TABLE test;  CREATE TYPE vector IS VARRAY(15) OF NUMBER;  DECLARE  TYPE tab\_index IS TABLE OF NUMBER  INDEX BY BINARY\_INTEGER;  v\_tab\_index tab\_index;  v\_tab\_imbri tab\_imbri;  v\_vector vector;  i INTEGER;  BEGIN  v\_tab\_index(1) := 72;  v\_tab\_index(2) := 23;  v\_tab\_imbri := tab\_imbri(5, 3, 2, 8, 7);  v\_vector := vector(1, 2);  i:=v\_tab\_index.FIRST;  DBMS\_OUTPUT.*PUT\_LINE*('v\_tab\_index: '||v\_tab\_index(i));  LOOP  i:= v\_tab\_index.*NEXT*(i);  DBMS\_OUTPUT.*PUT\_LINE*('v\_tab\_index: '||v\_tab\_index(i));  EXIT WHEN (i >= v\_tab\_index.LAST);  END LOOP;  i:=v\_tab\_imbri.FIRST;  WHILE (i <= v\_tab\_imbri.LAST) LOOP  DBMS\_OUTPUT.*PUT\_LINE*('v\_tab\_imbri: '||v\_tab\_imbri(i));  i:= v\_tab\_imbri.*NEXT*(i);  END LOOP;  FOR i IN 1..v\_vector.COUNT LOOP  DBMS\_OUTPUT.*PUT\_LINE*('vector: '||v\_vector(i));  END LOOP;  END;  /  */\* Lab 2 PLSQL ex. 1 edit solutie Mindruta Andrei\*/*  *DECLARE*  *--TYPE emp\_ids\_table IS TABLE OF EMPLOYEES.EMPLOYEE\_ID%TYPE;*  *TYPE emp\_ids\_table IS VARRAY (10) OF EMPLOYEES.EMPLOYEE\_ID%TYPE;*  *top\_emp\_ids emp\_ids\_table:=emp\_ids\_table();*  *BEGIN*  */\*SELECT EMPLOYEE\_ID*  *BULK COLLECT INTO top\_emp\_ids*  *FROM (SELECT \* FROM EMPLOYEES*  *ORDER BY SALARY DESC)*  *WHERE ROWNUM<=5;\*/*  *--FETCH FIRST 5 ROWS ONLY;*  *top\_emp\_ids.extend;*  *top\_emp\_ids(top\_emp\_ids.FIRST):=1;*  *FOR i IN 1 .. top\_emp\_ids.COUNT LOOP*  *DBMS\_OUTPUT.PUT\_LINE('Employee ID: ' || top\_emp\_ids(i));*  *END LOOP;*  *END;*  */*  --ne-am oprit la colectii pe mai multe niveluri |
| --- |

#### 12.11.2024, Laborator2\_PLSQL.pdf & Laborator3\_PLSQL.pdf

##### 

| SET SERVEROUTPUT ON  DROP TABLE job\_emp\_spr;  DROP TYPE list\_ang\_spr;  CREATE OR REPLACE TYPE list\_ang\_spr IS TABLE OF NUMBER(4);  /  CREATE TABLE job\_emp\_spr (  cod\_job VARCHAR2(10),  titlu\_job VARCHAR2(25),  info list\_ang\_spr)  NESTED TABLE info STORE AS info\_tab\_spr;  DECLARE  v\_list list\_ang\_spr := list\_ang\_spr (1002, 1003, 1004);  v\_info\_list list\_ang\_spr := list\_ang\_spr (1005);  v\_info job\_emp\_spr.info%TYPE;  v\_cod job\_emp\_spr.cod\_job%TYPE := 'IT\_PROG';  i INTEGER;  BEGIN  INSERT INTO job\_emp\_spr  VALUES ('AD\_PRES', 'Director', list\_ang\_spr (1000, 1001));  INSERT INTO job\_emp\_spr  VALUES ('IT\_PROG', 'Programator', v\_list);  INSERT INTO job\_emp\_spr  VALUES ('SA\_REP', 'Reprezentant vanzari', v\_info\_list);    SELECT info  INTO v\_info  FROM job\_emp\_spr  WHERE cod\_job = v\_cod;    --afisare v\_info  DBMS\_OUTPUT.PUT\_LINE('v\_info:');  i := v\_info.FIRST;  WHILE (i <= v\_info.last) LOOP  DBMS\_OUTPUT.PUT\_LINE(v\_info(i));  i := v\_info.next(i);  END LOOP;  END;  /  SELECT \*  FROM job\_emp\_spr;  SELECT \*  FROM TABLE (  SELECT info  FROM job\_emp\_spr  WHERE cod\_job = 'IT\_PROG');  -- urmeaza comenzi care merg numai pt col de tip tab imb!!!!  --adaugare element in colectie  INSERT INTO TABLE (  SELECT info  FROM job\_emp\_spr  WHERE cod\_job = 'IT\_PROG')  VALUES (0);  --adaugare elemente obtinute cu subcerere  INSERT INTO TABLE (  SELECT info  FROM job\_emp\_spr  WHERE cod\_job = 'IT\_PROG')  SELECT employee\_id  FROM employees  WHERE job\_id = 'IT\_PROG';  --modificare element colectie  UPDATE TABLE (  SELECT info  FROM job\_emp\_spr  WHERE cod\_job = 'IT\_PROG') a  SET VALUE(a) = 1006  WHERE COLUMN\_VALUE = 0;  --stergere element colectie  DELETE FROM TABLE (  SELECT info  FROM job\_emp\_spr  WHERE cod\_job = 'IT\_PROG') a  WHERE COLUMN\_VALUE = 1006;  CREATE OR REPLACE TYPE list\_ang\_spr IS VARRAY(10) OF NUMBER(4);  /  CREATE TABLE job\_emp\_spr (  cod\_job VARCHAR2(10),  titlu\_job VARCHAR2(25),  info list\_ang\_spr);  DECLARE  v\_list list\_ang\_spr := list\_ang\_spr (1002, 1003, 1004);  v\_info\_list list\_ang\_spr := list\_ang\_spr (1005);  v\_info job\_emp\_spr.info%TYPE;  v\_cod job\_emp\_spr.cod\_job%TYPE := 'IT\_PROG';  i INTEGER;  BEGIN  INSERT INTO job\_emp\_spr  VALUES ('AD\_PRES', 'Director', list\_ang\_spr (1000, 1001));  INSERT INTO job\_emp\_spr  VALUES ('IT\_PROG', 'Programator', v\_list);  INSERT INTO job\_emp\_spr  VALUES ('SA\_REP', 'Reprezentant vanzari', v\_info\_list);    SELECT info  INTO v\_info  FROM job\_emp\_spr  WHERE cod\_job = v\_cod;    --afisare v\_info  DBMS\_OUTPUT.PUT\_LINE('v\_info:');  i := v\_info.FIRST;  WHILE (i <= v\_info.last) LOOP  DBMS\_OUTPUT.PUT\_LINE(v\_info(i));  i := v\_info.next(i);  END LOOP;  END;  /  SELECT \*  FROM job\_emp\_spr;  SELECT \*  FROM TABLE (  SELECT info  FROM job\_emp\_spr  WHERE cod\_job = 'IT\_PROG');  -- urmeaza comenzi nu merg pt col de tip vector!!!!  --adaugare element in colectie  INSERT INTO TABLE (  SELECT info  FROM job\_emp\_spr  WHERE cod\_job = 'IT\_PROG')  VALUES (0);  --adaugare elemente obtinute cu subcerere  INSERT INTO TABLE (  SELECT info  FROM job\_emp\_spr  WHERE cod\_job = 'IT\_PROG')  SELECT employee\_id  FROM employees  WHERE job\_id = 'IT\_PROG';  --modificare element colectie  UPDATE TABLE (  SELECT info  FROM job\_emp\_spr  WHERE cod\_job = 'IT\_PROG') a  SET VALUE(a) = 1006  WHERE COLUMN\_VALUE = 0;  --stergere element colectie  DELETE FROM TABLE (  SELECT info  FROM job\_emp\_spr  WHERE cod\_job = 'IT\_PROG') a  WHERE COLUMN\_VALUE = 1006; |
| --- |

##### Pentru adaugare enunțuri și rezolvări utilizați sablonul:

##### Activitate: Completati codul lipsa din lab 2, exc 20 -> 10:50

##### Rezolvati

* exc 1 din lab 3 -> 11:15
* exc 4,
* exc 9 folosind 2 cursoare clasice cu WHILE, 2 ciclu cursoare, 2 ciclu cursoare cu subcereri, o expresie cursor (a se vedea exc 3 si 14) -> 11:50

Raman cursoarele dinamice si for update pt urm lab plus interpretare raspunsuri ex 9.

Tema de reflectie: ilustrati diferente intre cursoare pe schema hr…

#### 26.11.2024, Laborator3\_PLSQL.pdf si Laborator4\_PLSQL.pdf

##### 

| SET SERVEROUTPUT ON  SELECT \*  FROM USER\_OBJECTS  WHERE OBJECT\_TYPE IN ('PROCEDURE','FUNCTION');  SELECT \*  FROM USER\_SOURCE  WHERE TYPE IN ('PROCEDURE','FUNCTION');  SELECT \*  FROM USER\_ERRORS;  ALTER TABLE dept\_spr  DROP COLUMN info;  DECLARE  PROCEDURE add\_dept  (p\_cod dept\_spr.department\_id %TYPE,  p\_nume dept\_spr.department\_name %TYPE,  p\_manager dept\_spr.manager\_id %TYPE,  p\_location dept\_spr.location\_id%TYPE)  IS  BEGIN  INSERT INTO dept\_spr  VALUES (p\_cod, p\_nume, p\_manager,p\_location);  END;  BEGIN  add\_dept(45, 'DB Administration' , 100, 2700);  END;  /  SELECT \*  FROM dept\_spr;  ROLLBACK;  CREATE OR REPLACE PROCEDURE first\_spr  IS  azi DATE := SYSDATE;  ieri azi%TYPE;  BEGIN    DBMS\_OUTPUT.PUT\_LINE('Programare PL/SQL') ;  DBMS\_OUTPUT.PUT\_LINE(TO\_CHAR(azi, 'dd-month-yyyy hh24:mi:ss'));  ieri := azi -1;  DBMS\_OUTPUT.PUT\_LINE(TO\_CHAR(ieri, 'dd-mon-yyyy'));  END;  /  EXECUTE first\_spr  BEGIN  first\_spr;  END;  /  CREATE OR REPLACE FUNCTION p14l4\_pnu (p\_dept employees.department\_id%TYPE)  RETURN NUMBER IS  rezultat NUMBER;  BEGIN  SELECT COUNT(\*)  INTO rezultat  FROM employees  WHERE department\_id = p\_dept  AND TO\_CHAR(hire\_date,'yyyy') > 1995;  RETURN rezultat;  END p14l4\_pnu;  /  --a)  VARIABLE nr NUMBER  EXECUTE :nr := p14l4\_pnu (80);  PRINT nr  --b)  VARIABLE nr NUMBER  CALL p14l4\_pnu (50) INTO :nr;  PRINT nr  --c)  SELECT p14l4\_pnu (80)  FROM dual;  --d)  UNDEFINE p\_dep  DEFINE p\_dep = 50  -- sau ACCEPT p\_dep PROMPT 'Introduceti codul departamentului '  DECLARE  nr NUMBER;  v\_dep employees.department\_id%TYPE := &p\_dep;  BEGIN  nr := p14l4\_pnu (v\_dep);  IF nr <> 0 THEN  DBMS\_OUTPUT.PUT\_LINE('numarul salariatilor angajati dupa 1995 in  departamentul '||v\_dep || ' este '||nr);  ELSE  DBMS\_OUTPUT.PUT\_LINE('departamentul cu numarul '|| v\_dep || ' nu are  angajati');  END IF;  END;  /  --e)  EXEC DBMS\_OUTPUT.PUT\_LINE(p14l4\_pnu(80)) |
| --- |

##### 

##### Activitate: EX 5-7

#### 26.11.2024

##### nu s-a tinut

#### 10.12.2024

##### consultatii proiect

#### 7.01.2024

##### nu s-a tinut, zi libera

##### End