PROIECT - SISTEME DE GESTIUNE A BAZELOR DE DATE

-Symphony MUSIC SCHOOL-

Nume: Ion Melania-Victorița

Grupa: 241

PROIECT

SISTEME DE GESTIUNE A BAZELOR DE DATE

Exercitiul 1: Prezentați pe scurt baza de date (utilitatea ei).

O scoală privată de arte pregătește o multitudine de cursanți pentru a excela în domeniu. Școala oferă diverse cursuri precum canto, chitară, pian, vioară sau producție muzicală. Orele se desfășoară individual și urmăresc atingerea obiectivelor fiecărui cursant.

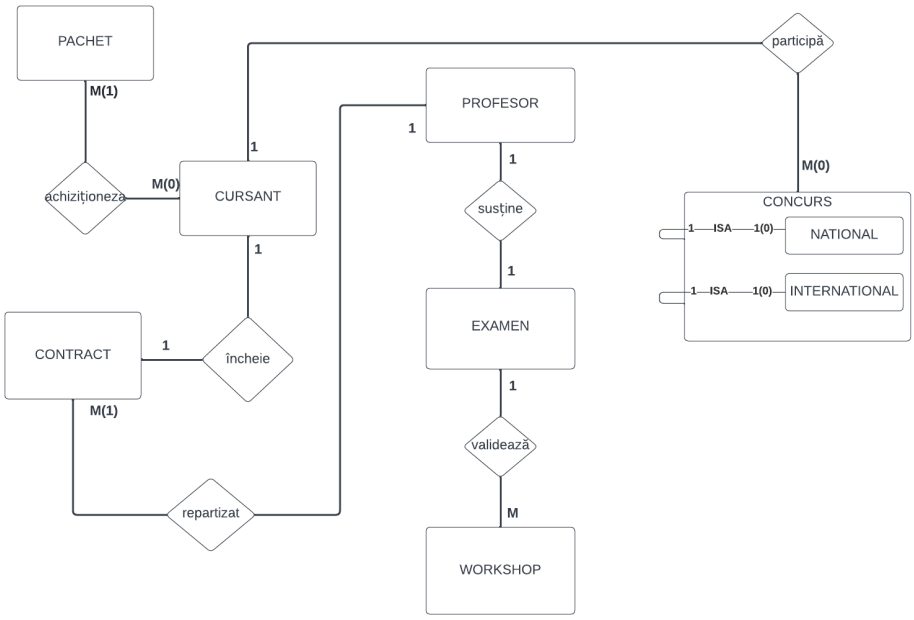
La venirea unui nou cursant, acesta participă gratuit la un test de aptitudini pentru a i se evalua abilitățile.Elevii care sunt admiși au obținut o notă mai mare sau egală cu 6 și li se creează un contract si sunt repartizati unui profesor. În plus, celor care obțin o notă de admitere între 9 si 10 li se va adăuga si o bursă de 200 lei.

Cursanții pot achizitiona un pachet(abonament) sau mai multe în functie de ce vor dori sa studieze. Pachetul va avea o dată de început si una de finalizare, iar acesta trebuie achitat până la data finalizării. Pachetele se vor desfășura pe o perioadă de o lună de zile în care cursantul poate participa la un anumit număr de ore (4/8 ore pe lună) în functie de preferințe. Astfel, ele sunt de două tipuri: basic sau premium. Modalitățile de plată sunt cash sau card, iar pentru plata prin card se va oferi o reducere de 5%/pachet.

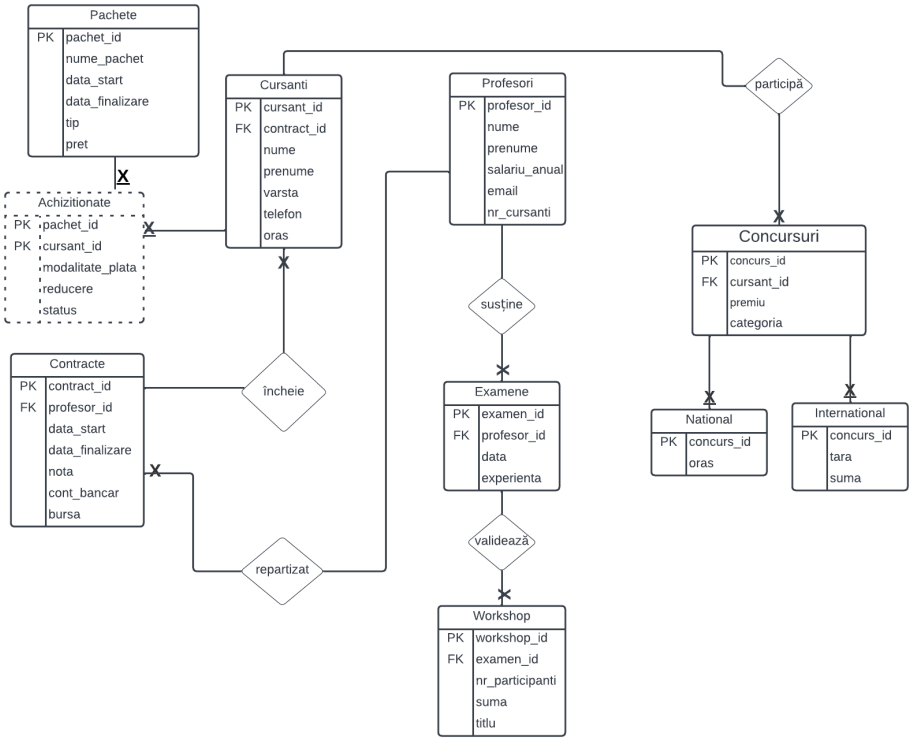
Profesorul își poate pregăti și înscrie cursantul la diverse concursuri în funcție de preferințele și nivelul elevului. Concursurile pot fi naționale și internaționale, pe anumite categorii. Pentru obținere premiului I la un concurs internatațional, cursantul va primi o bursă de 500 lei.

Pentru a selecta cei mai buni profesori, în momentul angajării vor susține un examen de competențe, unde vor mentiona si experienta in domeniu. Profesorii cu o anumită experiență pot fi aleși de către conducere să organizeze diverse workshopuri gratuite (pot participa persoane din exteriorul școlii) pentru care vor fi platiti in plus.

Exercitiul 2: Realizați diagrama entitate-relație (ERD).



Exercitiul 3: Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare.



Exercitiul 4: Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

CREATE TABLE cursanti(cursant_id number(10) NOT NULL, contract_id number(10), nume varchar2(50) NOT NULL, prenume varchar2(50), varsta number(2), telefon varchar2(10) NOT NULL, oras varchar2(50) DEFAULT 'Bucuresti', CONSTRAINT cursanti_pk PRIMARY KEY(cursant_id), CONSTRAINT cursanti_fk FOREIGN KEY(contract_id) REFERENCES contracte);

CREATE TABLE profesori(profesor_id number(10) NOT NULL,cursant_id number(10), nume varchar2(50) NOT NULL, prenume varchar2(50), salariu_anual number(6) DEFAULT 3000 not null, email varchar2(20) not null, nr_cursanti number(6) NOT NULL, CONSTRAINT profesori_pk PRIMARY KEY(profesor_id));

CREATE TABLE examene(examen_id number(10) NOT NULL, profesor_id number(10), data date NOT NULL, experienta number(2),CONSTRAINT examene_pk PRIMARY KEY(examen_id), CONSTRAINT examene_fk FOREIGN KEY(profesor_id) REFERENCES profesori);

CREATE TABLE contracte(contract_id number(10) NOT NULL, profesor_id number(10), data_start date NOT NULL, data_finalizare date, nota float(3) NOT NULL, bursa number(5,2) DEFAULT 0, cont_bancar varchar2(16),CONSTRAINT contracte_pk PRIMARY KEY(contract_id));

CREATE TABLE workshop(workshop_id number(10) NOT NULL, examen_id number(10), nr_participanti number(5) DEFAULT 0, titlu varchar2(50) UNIQUE, suma number(4) DEFAULT(100),CONSTRAINT workshop_pk PRIMARY KEY(workshop_id),CONSTRAINT workshop_fk FOREIGN KEY(examen_id) REFERENCES examene);

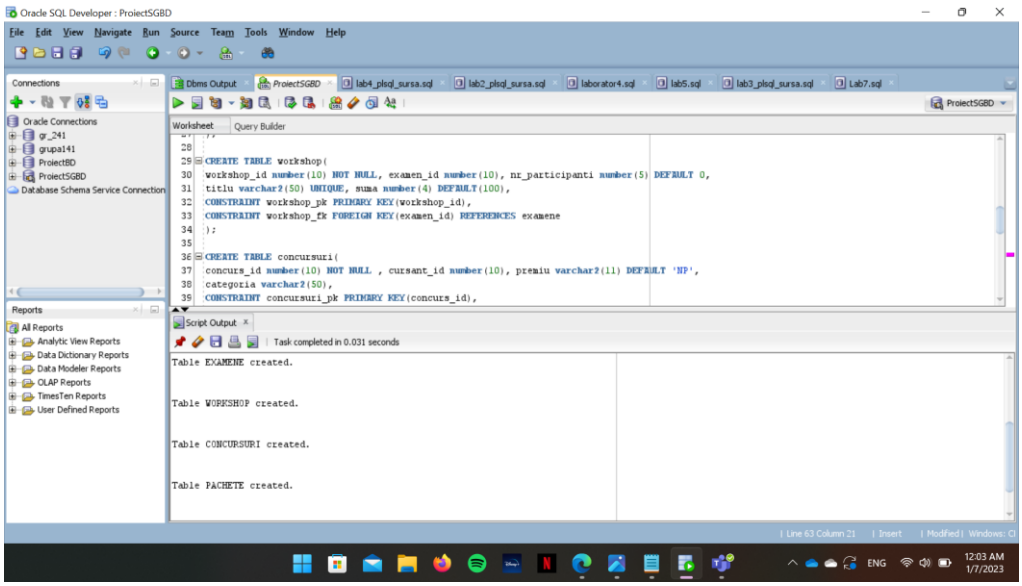
CREATE TABLE concursuri(concurs_id number(10) NOT NULL , cursant_id number(10), premiu varchar2(11) DEFAULT 'NP',categoria varchar2(50), CONSTRAINT concursuri_pk PRIMARY KEY(concurs_id),CONSTRAINT concursuri_fk FOREIGN KEY(cursant_id) REFERENCES cursanti);

CREATE TABLE international(concurs_id number(10) NOT NULL, tara varchar2(40) NOT NULL, suma number(5) DEFAULT 0,CONSTRAINT international_pk PRIMARY KEY(concurs_id));

CREATE TABLE national(concurs_id number(10) NOT NULL, oras varchar2(50) DEFAULT 'Bucuresti',CONSTRAINT national_pk PRIMARY KEY(concurs_id));

CREATE TABLE pachete(pachet_id number(10) NOT NULL, nume_pachet varchar2(30), data_start date NOT NULL, data_finalizare date NOT NULL, tip varchar2(20) NOT NULL, pret number(4) NOT NULL,CONSTRAINT pachete_pk PRIMARY KEY(pachet_id));

CREATE TABLE achizitionate(pachet_id number(10) NOT NULL, cursant_id number(10) NOT NULL,modalitate_plata varchar2(10) DEFAULT 'cash',reducere varchar2(4), status varchar2(10) NOT NULL, CONSTRAINT achizitionate_pk PRIMARY KEY(pachet_id,cursant_id));



Exercitiul 5: Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

```
INSERT INTO contracte
VALUES(1,1,DATE '2021-10-5',DATE '2022-6-18',6,0,'RO60PORL6419564');
commit;
INSERT INTO contracte
VALUES(2,2,DATE '2021-12-23', DATE '2022-11-25',7.90,0,'RO60PORL6439521');
commit;
 INSERT INTO contracte
VALUES(3,3,DATE '2021-11-12',NULL,9,200,'RO60PORL6419564');
commit;
INSERT INTO contracte
VALUES(4,3, DATE '2021-12-25',NULL,9,700,'RO60PORL6419569');
commit;
INSERT INTO contracte
VALUES(5,1,DATE '2021-10-11', DATE '2022-12-22',8,500,'RO60PORL64146721');
commit;
```

```
INSERT INTO contracte
VALUES(6,2,DATE '2020-11-6',NULL,6,0,'RO60PORL6419710');
commit;
INSERT INTO contracte
VALUES(7,4,DATE '2021-6-5',NULL,7,0,'RO60PORL6419723');
commit;
INSERT INTO contracte
VALUES(8,5,DATE '2019-11-13',NULL,10,200,'RO60PORL6419754');
commit;
INSERT INTO contracte
VALUES(9,3,DATE '2019-9-12',NULL,7,0,'RO60PORL6419788');
commit;
INSERT INTO contracte
VALUES(10,1,DATE '2019-12-10',NULL,8,0,'RO60PORL6419788');
commit;
 INSERT INTO profesori
VALUES(1,'Campbell','Paula',15000,'paula@test.com',2);
commit;
INSERT INTO profesori
VALUES(2,'Arnold','Lenore',25000,'lenore@test.com',1);
commit;
INSERT INTO profesori
VALUES(3,'Miller','Francesca',12500,'francesca@test.com',3);
commit;
INSERT INTO profesori
VALUES(4,'Carson','Angie',37000,'angie@test.com',1);
commit;
```

```
INSERT INTO profesori
VALUES(5,'Castillo','Camilla',50000,'camilla@test.com',1);
commit;
 INSERT INTO cursanti
VALUES(1,10,'Thorne','Ariella',15,'0735469834','Brasov');
commit;
INSERT INTO cursanti
VALUES(2,9,'Crossby', 'Tessa', 13, '0756289576', 'Sibiu');
commit;
INSERT INTO cursanti
VALUES(3,8,'Palmer','Penelope',18,'0725748954','Oradea');
commit;
INSERT INTO cursanti
VALUES(4,7,'Davison','Sofia',20,'0724598378','Brasov');
commit;
INSERT INTO cursanti
VALUES(5,6,'Steveson', 'Becky', 16, '0769837568', 'Sibiu');
commit;
INSERT INTO cursanti
VALUES(6,5,'Popescu','Ioana',9,'0735469811','Brasov');
commit;
INSERT INTO cursanti
VALUES(7,4,'Ion','Madalin',11,'0735469832','Bucuresti');
commit;
INSERT INTO cursanti
VALUES(8,3,'Serban','Maria',16,'0735789838','Bucuresti');
commit;
```

```
INSERT INTO cursanti
VALUES(9,2,'Radu','Amalia',17,'0745469839','Bucuresti');
commit;
INSERT INTO cursanti
VALUES(10,1,'Emil','Alex',15,'0735129830','Brasov');
commit;

CREATE SEQUENCE SEQ_INSERT2 START WITH 10 INCREMENT BY 1;

INSERT INTO pachete
VALUES(SEQ_INSERT2.NEXTVAL,'canto',DATE '2022-11-4',DATE '2022-12-4','premium','600');
commit;
INSERT INTO pachete
VALUES(SEQ_INSERT2.NEXTVAL,'canto',DATE '2022-12-14',DATE '2023-1-14','basic','300');
commit;
INSERT INTO pachete
VALUES(SEQ_INSERT2.NEXTVAL,'pian',DATE '2022-12-4',DATE '2023-1-4','premium','400');
commit;
INSERT INTO pachete
VALUES(SEQ_INSERT2.NEXTVAL,'pian',DATE '2022-11-13',DATE '2022-12-13','basic','300');
commit;
INSERT INTO pachete
VALUES(SEQ_INSERT2.NEXTVAL,'chitara',DATE '2022-10-14',DATE '2022-10-14','premium','300');
commit;
INSERT INTO pachete
VALUES(SEQ_INSERT2.NEXTVAL,'chitara',DATE '2022-12-18',DATE '2023-1-18','basic','200');
commit;
```

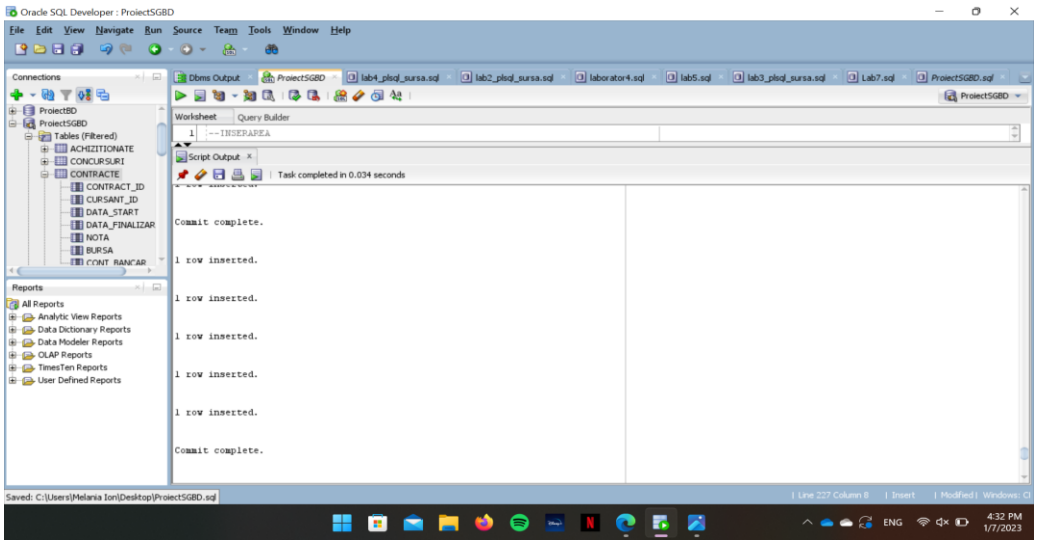
```
INSERT INTO pachete
VALUES(SEQ_INSERT2.NEXTVAL,'canto',DATE '2022-11-29',DATE '2022-12-29','premium','550');
commit;
INSERT INTO pachete
VALUES(SEQ_INSERT2.NEXTVAL,'canto',DATE '2022-9-4',DATE '2022-10-4','basic','400');
commit;
INSERT INTO pachete
VALUES(SEQ_INSERT2.NEXTVAL,'canto',DATE '2022-8-8',DATE '2022-9-8','premium','550');
commit;
INSERT INTO pachete
VALUES(SEQ_INSERT2.NEXTVAL,'canto',DATE '2022-12-25',DATE '2023-1-25','basic','400');
commit;
 INSERT INTO achizitionate
VALUES(10,1,'card','5%','achitat');
commit;
INSERT INTO achizitionate
VALUES(11,2,'card','5%','neachitat');
commit;
INSERT INTO achizitionate
VALUES(12,3,'card','5%','achitat');
commit;
INSERT INTO achizitionate
VALUES(13,4,'card','5%','achitat');
commit;
INSERT INTO achizitionate
VALUES(14,5,'card','5%','achitat');
commit;
```

```
INSERT INTO achizitionate
VALUES(15,6,'cash','0%','neachitat');
commit;
INSERT INTO achizitionate
VALUES(16,7,'cash','0%','achitat');
commit;
INSERT INTO achizitionate
VALUES(17,8,'cash','0%','achitat');
commit;
INSERT INTO achizitionate
VALUES(18,9,'cash','0%','achitat');
commit;
INSERT INTO achizitionate
VALUES(19,10,'cash','0%','neachitat');
commit;
 INSERT INTO examene
VALUES(1,2,DATE '2019-3-5',5);
commit;
INSERT INTO examene
VALUES(2,3,DATE '2019-6-20',10);
commit;
INSERT INTO examene
VALUES(3,1,DATE '2019-07-8',3);
commit;
INSERT INTO examene
VALUES(4,5,DATE '2021-03-15',2);
commit;
```

```
INSERT INTO examene
VALUES(5,4,DATE '2020-02-19',9);
commit;

CREATE SEQUENCE SEQ_INSERT3 START WITH 33 INCREMENT BY 10;
INSERT INTO workshop
VALUES(SEQ_INSERT3.NEXTVAL,1,30,'The truth about singing',150);
commit;
INSERT INTO workshop
VALUES(SEQ_INSERT3.NEXTVAL,2,100,'How to become a better pianist',550);
commit;
INSERT INTO workshop
VALUES(SEQ_INSERT3.NEXTVAL,2,300,'About guitars',900);
commit;
INSERT INTO workshop
VALUES(SEQ_INSERT3.NEXTVAL,5,45,'How to sing whistle notes',350);
commit;
INSERT INTO workshop
VALUES(SEQ_INSERT3.NEXTVAL,5,70,'How to write a song',400);
commit;
 CREATE SEQUENCE SEQ_INSERT4 START WITH 1 INCREMENT BY 1;
 INSERT INTO concursuri
VALUES(SEQ_INSERT4.NEXTVAL,10,'mentiune','chitara');
INSERT INTO concursuri
VALUES(SEQ_INSERT4.NEXTVAL,9,'mentiune','chitara');
INSERT INTO concursuri
VALUES(SEQ_INSERT4.NEXTVAL,8,'II','canto');
```

```
INSERT INTO concursuri
VALUES(SEQ_INSERT4.NEXTVAL,7,'I','canto');
INSERT INTO concursuri
VALUES(SEQ_INSERT4.NEXTVAL,6,'I','pian');
INSERT INTO concursuri
VALUES(SEQ_INSERT4.NEXTVAL,1,'mentiune','pian');
INSERT INTO concursuri
VALUES(SEQ_INSERT4.NEXTVAL,2,'III','chitara');
INSERT INTO concursuri
VALUES(SEQ_INSERT4.NEXTVAL,3,'II','chitara');
INSERT INTO concursuri
VALUES(SEQ_INSERT4.NEXTVAL,4,'II','canto');
INSERT INTO concursuri
VALUES(SEQ_INSERT4.NEXTVAL,5,'mentiune','canto');
commit;
 INSERT INTO national
VALUES(1,'Sibiu');
INSERT INTO national
VALUES(2,'Brasov');
INSERT INTO national
VALUES(3,'Bucuresti');
INSERT INTO national
VALUES(6,'Bucuresti');
INSERT INTO national
VALUES(7,'Sibiu');
commit;
```

```
INSERT INTO international
VALUES(4,'Spania',1500);
INSERT INTO international
VALUES(5,'Canada',4000);
INSERT INTO international
VALUES(8,'Italia',0);
INSERT INTO international
VALUES(9,'Franta',700);
INSERT INTO international
VALUES(10,'Olanda',460);
commit;
```



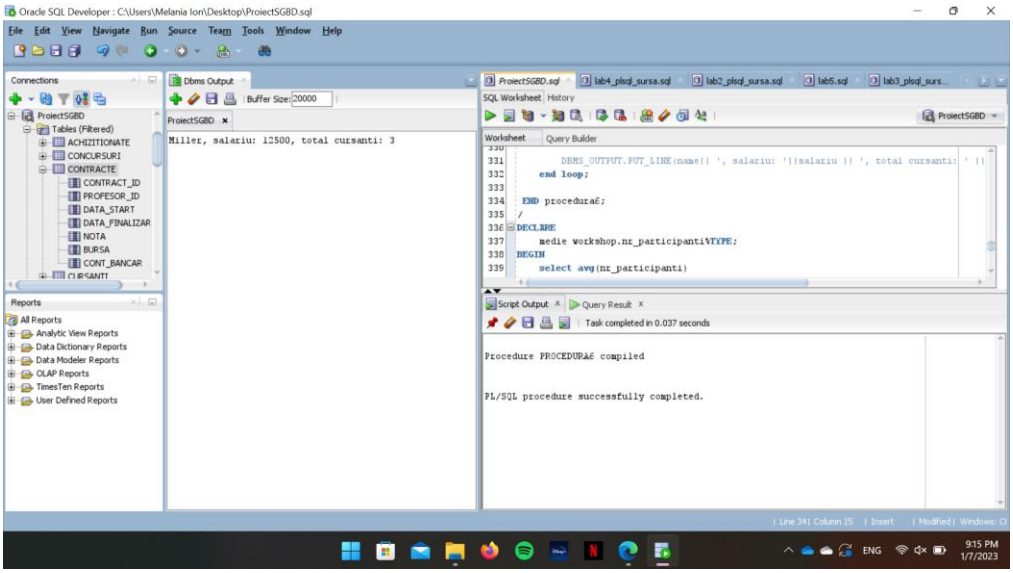
Exercitiul 6: Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze două tipuri diferite de colecții studiate. Apelați subprogramul.

CERINȚĂ: Pentru workshop-urile care au avut numarul participantilor mai mare decat media, afisati numele, salariul si numarul de cursanti al profesorilor care au tinut acel workshop.

REZOLVARE:

```
CREATE OR REPLACE PROCEDURE procedura6

(v_medie workshop.nr_participanti%TYPE)

IS

TYPE myvector IS VARRAY(20) OF NUMBER;

  vector myvector:= myvector();

TYPE tablou_imbricat IS TABLE OF NUMBER;

timb tablou_imbricat := tablou_imbricat();

 salariu profesori.salariu_anual%TYPE;

name   profesori.nume%TYPE;

total profesori.nr_cursanti%TYPE;

BEGIN

   select examen_id

   bulk collect into vector       --id-urile examenelor

   from workshop

   where nr_participanti >= v_medie;
```

```
 for i in vector.first..vector.last loop
        timb.extend();
        select profesor_id
        into timb(i)       --id-urile profesorilor
        from examene
        where examen_id = vector(i);
    end loop;
   for i in timb.first..timb.last loop
        select nume,salariu_anual, nr_cursanti
        into name, salariu, total
        from profesori
        where profesor_id = timb(i);
         DBMS_OUTPUT.PUT_LINE(name|| ', salariu: '||salariu || ', total cursanti: ' || total);
    end loop;
END procedura6;
/
DECLARE
    medie workshop.nr_participanti%TYPE;
BEGIN
    select avg(nr_participanti)
    into medie
    from workshop;
   procedura6(medie);
END;
/
```



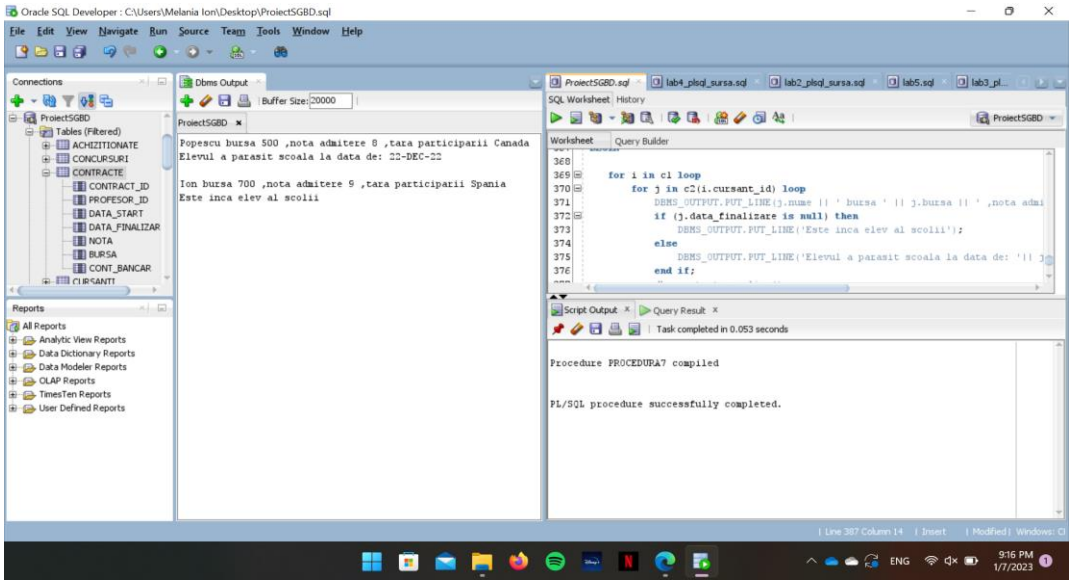
Exercitiul 7: Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat. Apelați subprogramul.

CERINȚĂ: Afisati numele cursantilor, bursa, nota pe care au primit-o la admitere, precum si daca mai sunt cursanti ai scolii sau nu in prezent, pentru cei care au participat la un concurs international si au obtinut premiul I doar la categoriile canto sau pian. Afisati in ordine crescatoare dupa tara in care a avut loc concursul.

REZOLVARE:

```
CREATE OR REPLACE PROCEDURE procedura7

(categ1 concursuri.categoria%type,categ2 concursuri.categoria%type)

IS

cursor c1 is

   select c.concurs_id, c.cursant_id, i.tara

   from concursuri c join international i on c.concurs_id = i.concurs_id

   where premiu = 'I' and categoria in (categ1,categ2)

   order by tara asc;

cursor c2 (cursant cursanti.cursant_id%type) is

   select p.nume, c.bursa, c.nota, c.data_finalizare

   from cursanti p join contracte c on p.contract_id = c.contract_id

   where p.cursant_id = cursant;
```

```
BEGIN
   for i in c1 loop
      for j in c2(i.cursant_id) loop
         DBMS_OUTPUT.PUT_LINE(j.nume || ' bursa ' || j.bursa || ' ,nota admitere ' || j.nota || ' ,tara participarii ' || i.tara);
         if (j.data_finalizare is null) then
            DBMS_OUTPUT.PUT_LINE('Este inca elev al scolii');
         else
            DBMS_OUTPUT.PUT_LINE('Elevul a parasit scoala la data de: '|| j.data_finalizare);
         end if;
         dbms_output.new_line();
      end loop;
   end loop;
END procedura7;
DECLARE
   categorie1 varchar2(10) := 'canto';
   categorie2 varchar2(10) := 'pian';
BEGIN
   procedura7(categorie1,categorie2);
END;
```



Exercitiul 8 : Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Definiți minim 2 excepții. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

CERINȚĂ: Pentru profesorii care au un salariu mai mare decat o valoare data, sa se calculeze suma totala a burselor cursantilor care sunt pregatiti de acesti profesori si care au o experienta mai mare de 5 ani. Erorile tratate:

-- nu exista profesori care sa aiba un astfel de salariu dat

-- valoarea data sa fie invalida

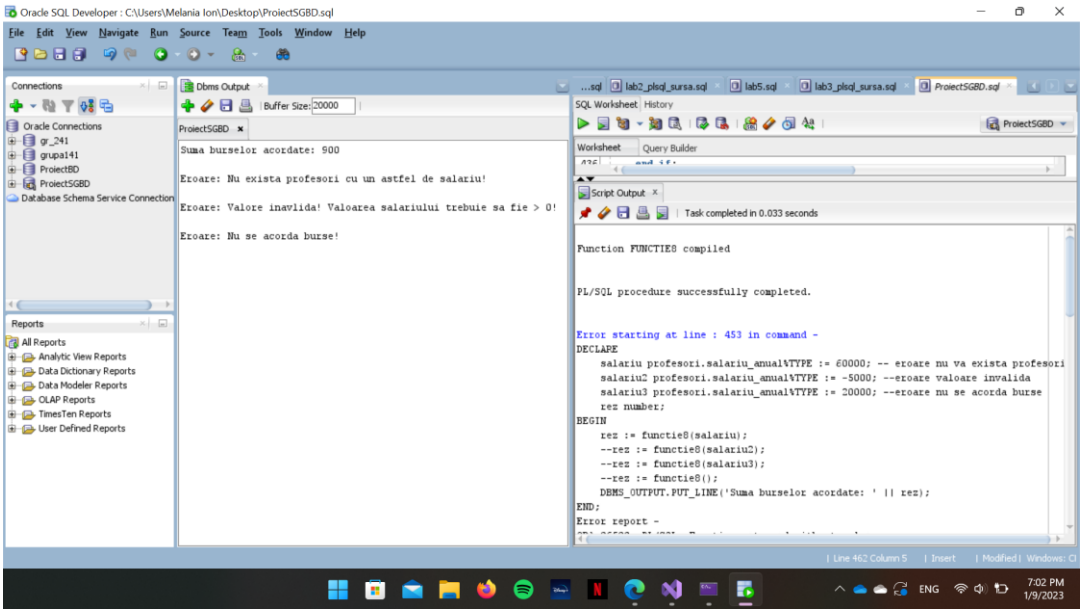
-- niciun cursant pentru profesorii care indeplinesc aceste conditii nu a obtinut bursa

REZOLVARE:

```
CREATE OR REPLACE FUNCTION functie8

( v_salariu profesori.salariu_anual%TYPE DEFAULT 10000)

RETURN number IS

v_total number := 0;

valoare_invalida exception;

exceptieP exception;

valid_bursa exception;

totalP number(3) := 0;
```

```
cursor cursor8 is
   select distinct c.contract_id ,p.profesor_id, p.nr_cursanti,p.salariu_anual, e.experienta, c.bursa
   from contracte c join profesori p on c.profesor_id = p.profesor_id
   join examene e on e.profesor_id = p.profesor_id
   where salariu_anual > v_salariu
   order by salariu_anual asc;
BEGIN
    if (v_salariu < 0) then
       raise valoare_invalida;
   end if;
   for i in cursor8 loop
       totalP := totalP + 1;
       if (i.experienta >= 5) then
          v_total := v_total + i.bursa;
       end if;
  end loop;
    if (totalP = 0 ) then
       raise exceptieP;
    end if;
   if (v_total = 0) then
       raise valid_bursa;
   end if;
   return v_total;
EXCEPTION
    WHEN valoare_invalida THEN
       DBMS_OUTPUT.PUT_LINE('Eroare: Valore inavlida! Valoarea salariului trebuie sa fie > 0!');
```

```
  WHEN exceptieP THEN
      DBMS_OUTPUT.PUT_LINE('Eroare: Nu exista profesori cu un astfel de salariu!');
   WHEN valid_bursa THEN
      DBMS_OUTPUT.PUT_LINE('Eroare: Nu se acorda burse!');
    WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Alta eroare!' ||  SQLERRM(SQLCODE));
END functie8;
/
DECLARE
   salariu profesori.salariu_anual%TYPE := 60000; -- eroare nu va exista profesori
   salariu2 profesori.salariu_anual%TYPE := -5000; --eroare valoare invalida
   salariu3 profesori.salariu_anual%TYPE := 20000; --eroare nu se acorda burse
   rez number;
BEGIN
   --rez := functie8(salariu);
   --rez := functie8(salariu2);
   --rez := functie8(salariu3);
   rez := functie8();
   DBMS_OUTPUT.PUT_LINE('Suma burselor acordate: ' || rez);
END;
/
```



Exercitiul 9: Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO_DATA_FOUND și TOO_MANY_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

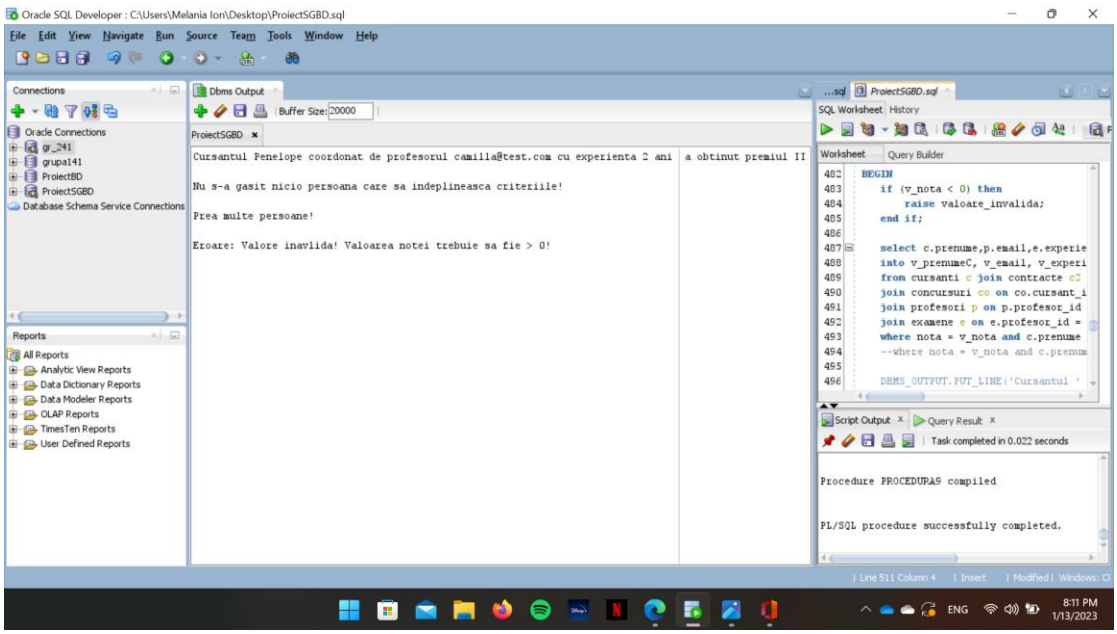
CERINȚĂ: Pentru un cursant care a obtinut la admitere o nota si al carui nume incepe cu o anumita litera (date), sa se afise numele profesorului care il pregateste, adresa sa de contact (email), precum si premiul obtinut de catre elev la un concurs.

REZOLVARE:

```
CREATE OR REPLACE PROCEDURE procedura9

(v_nota contracte.nota%type)

IS

   v_email profesori.email%type;

   v_prenumeC cursanti.prenume%type;

   v_experienta examene.experienta%type;

   v_premiu concursuri.premiu%type;

   valoare_invalida exception;
```

```
BEGIN
   if (v_nota < 0) then
      raise valoare_invalida;
   end if;
   select c.prenume,p.email,e.experienta,co.premiu
   into v_prenumeC, v_email, v_experienta, v_premiu
   from cursanti c join contracte c2 on c.contract_id = c2.contract_id
   join concursuri co on co.cursant_id = c.cursant_id
   join profesori p on p.profesor_id = c2.profesor_id
   join examene e on e.profesor_id = p.profesor_id
   where nota = v_nota and c.prenume like 'P%';
  --where nota = v_nota and c.prenume like 'M%'; --eroare
   DBMS_OUTPUT.PUT_LINE('Cursantul '  || v_prenumeC || ' coordonat de profesorul ' || v_email || '
cu experienta ' ||
   v_experienta || ' ani '|| ' a obtinut premiul ' ||v_premiu);
EXCEPTION
   WHEN valoare_invalida THEN
      DBMS_OUTPUT.PUT_LINE('Eroare: Valore inavlida! Valoarea notei trebuie sa fie > 0!');
   WHEN NO_DATA_FOUND THEN
       DBMS_OUTPUT.PUT_LINE('Nu s-a gasit nicio persoana care sa indeplineasca criteriile!');
   WHEN TOO_MANY_ROWS THEN
       DBMS_OUTPUT.PUT_LINE('Prea multe persoane!');
END procedura9;
DECLARE
  --afisare corecta:
  v_nota contracte.nota%type := 10; -- si nume care incepe cu litera P
```

```
  --eroare NO DATA FOUND
  --v_nota contracte.nota%type := 9; -- si nume care incepe cu litera P
  --eroare TOO MANY ROWS
  --v_nota contracte.nota%type := 9; -- si nume care incepe cu litera M
  --eroare Valoare Invalida
  --v_nota contracte.nota%type := -6;
BEGIN
   procedura9(v_nota);
END;
```



Exercitiul 10: Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

REZOLVARE: -S-a creat un trigger de tip LMD la nivel de comandă care sa actualizeze numarul de cursanti pentru tabelul profesori la orice comanda de INSERT, DELETE sau UPDATE asupra tabelului de contracte (intrucat un nou contract semnifica un nou cursant care va fi repartizat la un profesor).

```
CREATE OR REPLACE PROCEDURE actualizeaza_nr_cursanti
IS
BEGIN
   UPDATE profesori p
   SET nr_cursanti = (SELECT COUNT(*)
                      FROM contracte c
                      WHERE p.profesor_id = c.profesor_id);
END;
CREATE OR REPLACE TRIGGER trig10
AFTER INSERT OR DELETE OR UPDATE ON contracte
BEGIN
   IF INSERTING THEN
      actualizeaza_nr_cursanti();
      DBMS_OUTPUT.PUT_LINE('DUPA INSERARE: S-a actualizat si numarul cursantilor din tabelul
PROFESORI');
   ELSIF DELETING THEN
      actualizeaza_nr_cursanti();
      DBMS_OUTPUT.PUT_LINE('DUPA STERGERE:S-a actualizat si numarul cursantilor din tabelul
PROFESORI');
   ELSE
      actualizeaza_nr_cursanti();
      DBMS_OUTPUT.PUT_LINE('DUPA UPDATE: S-a actualizat si numarul cursantilor din tabelul
PROFESORI');
   END IF;
END;
/
ALTER TRIGGER trig10 ENABLE;
ALTER TRIGGER trig10 DISABLE;
```

--DECLANSARE TRIGGER 10

--inserare

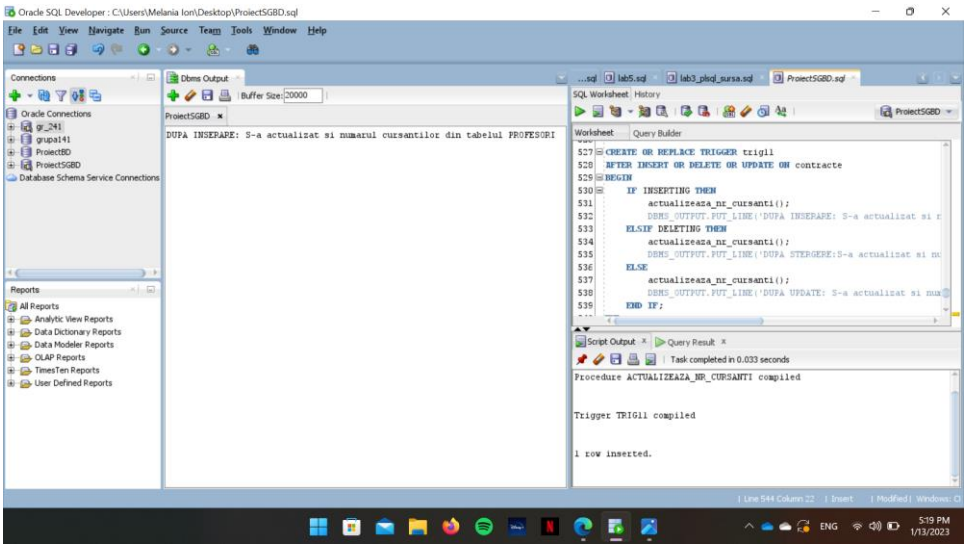
INSERT INTO contracte

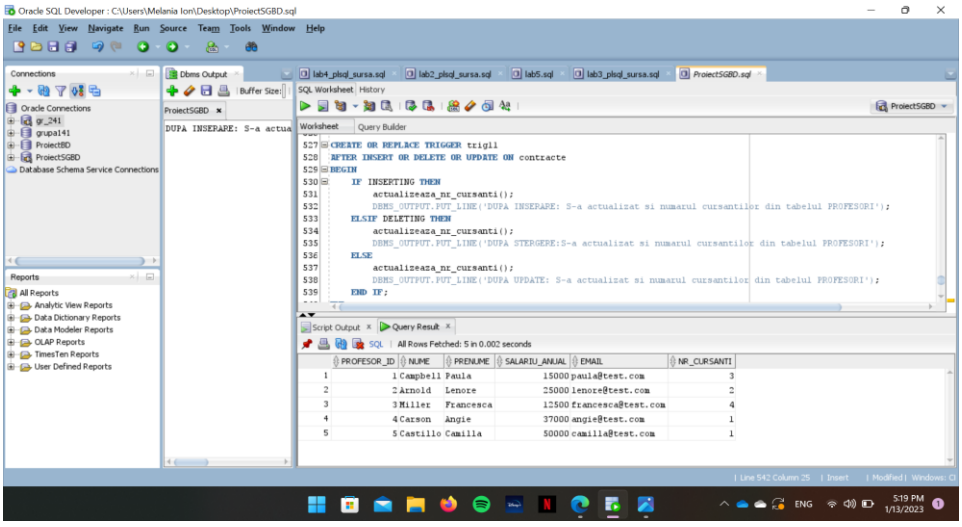
VALUES(11,3,SYSDATE,NULL,7,0,'RO60PORL3019546');

--stergere

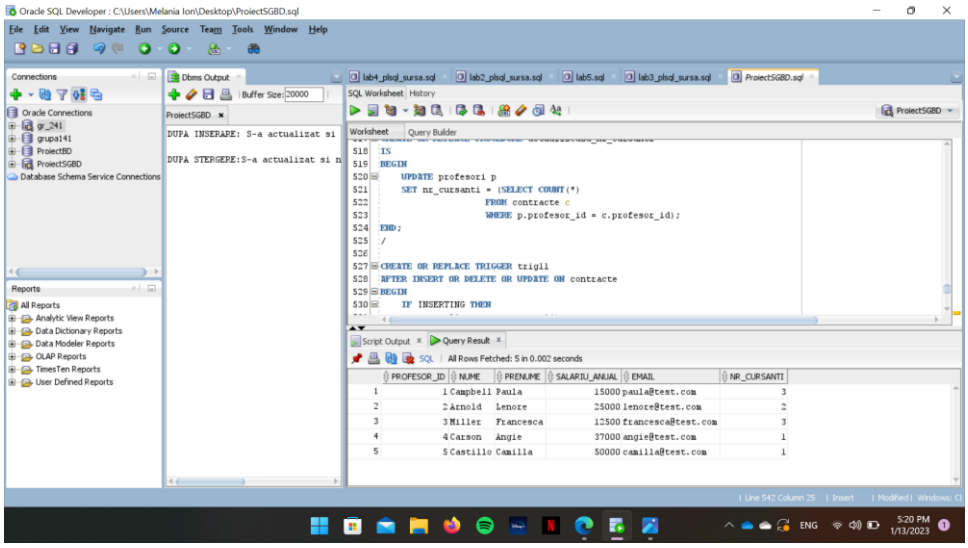
DELETE FROM contracte WHERE contract_id =11;

--Dupa comanda INSERT





--Dupa comanda DELETE



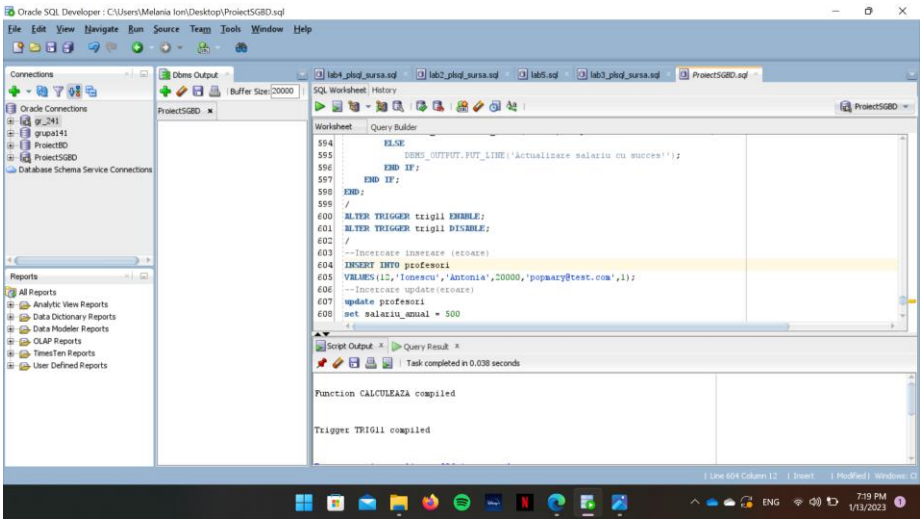
Exercitiul 11: Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.

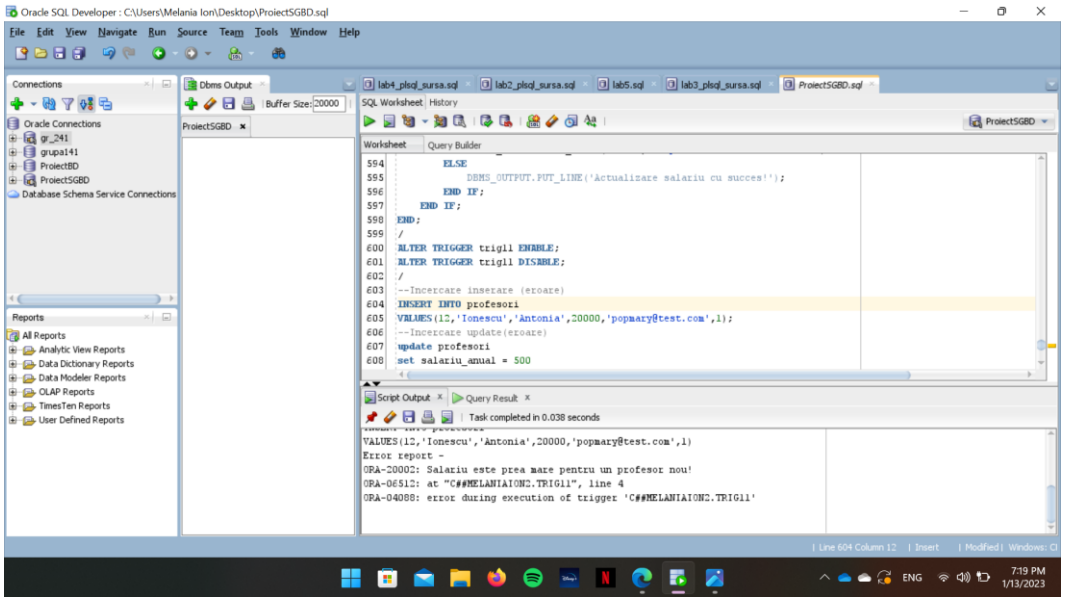
REZOLVARE: S-a creat un trigger de tip LMD la nivel de linie care nu permite inserarea unui profesor cu salariul mai mare decat media salariilor profesorilor care au mai mult de 2 cursanti, dar nici inserarea unui profesor cu un salariu mai mic de 12500. De asemenea nu se permite micsorarea salariilor. Daca se respecta aceste criterii, se va afisa un mesaj de succes al actiunii dorite.

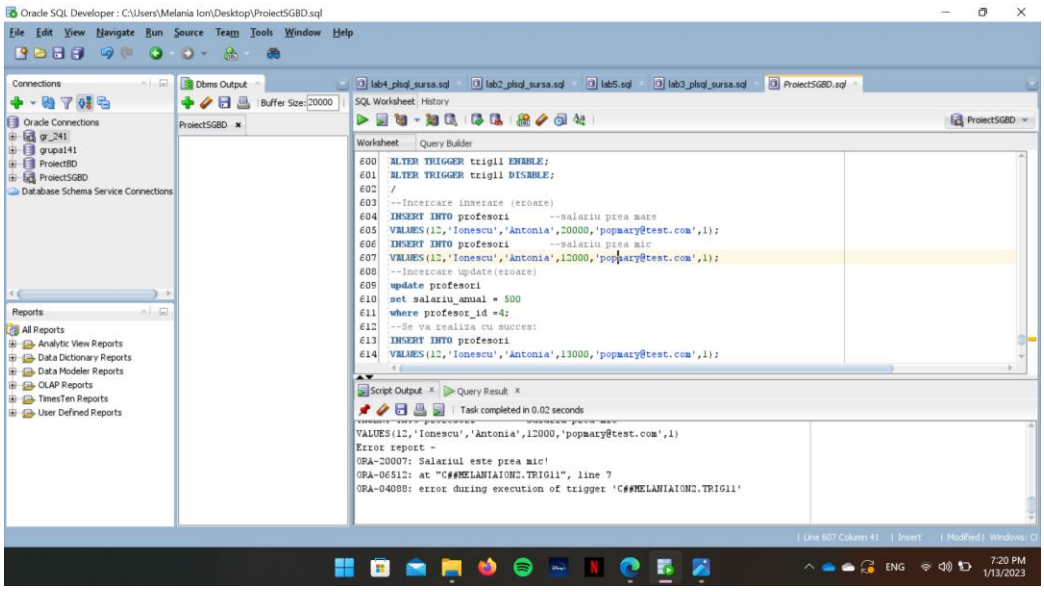
```
CREATE OR REPLACE FUNCTION calculeaza

RETURN number IS

   v_medie_salariu number (6);

BEGIN

   SELECT avg(salariu_anual)

   INTO v_medie_salariu

   FROM profesori

   where nr_cursanti > 2;

   return(v_medie_salariu);

END;

/
```

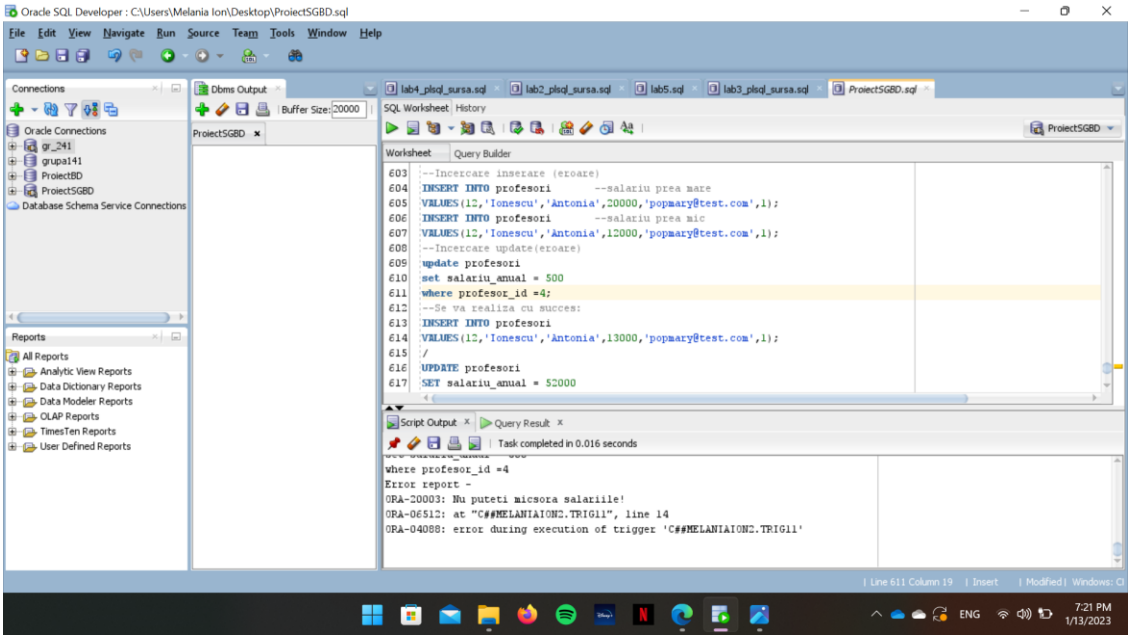
```
CREATE OR REPLACE TRIGGER trig11
BEFORE INSERT OR UPDATE ON profesori
FOR EACH ROW
BEGIN
   IF INSERTING THEN
      IF :NEW.salariu_anual > calculeaza() THEN
         RAISE_APPLICATION_ERROR(-20002,'Salariu este prea mare pentru un profesor nou!');
      ELSIF :NEW.salariu_anual < 12500 THEN
         RAISE_APPLICATION_ERROR(-20007,'Salariul este prea mic!');
      ELSE
         DBMS_OUTPUT.PUT_LINE('Inserare cu succes!');
      END IF;
   END IF;
   IF UPDATING THEN
      IF :NEW.salariu_anual < :OLD.salariu_anual THEN
         RAISE_APPLICATION_ERROR(-20003,'Nu puteti micsora salariile!');
      ELSE
         DBMS_OUTPUT.PUT_LINE('Actualizare salariu cu succes!');
      END IF;
   END IF;
END;
/
ALTER TRIGGER trig11 ENABLE;
ALTER TRIGGER trig11 DISABLE;
/
```

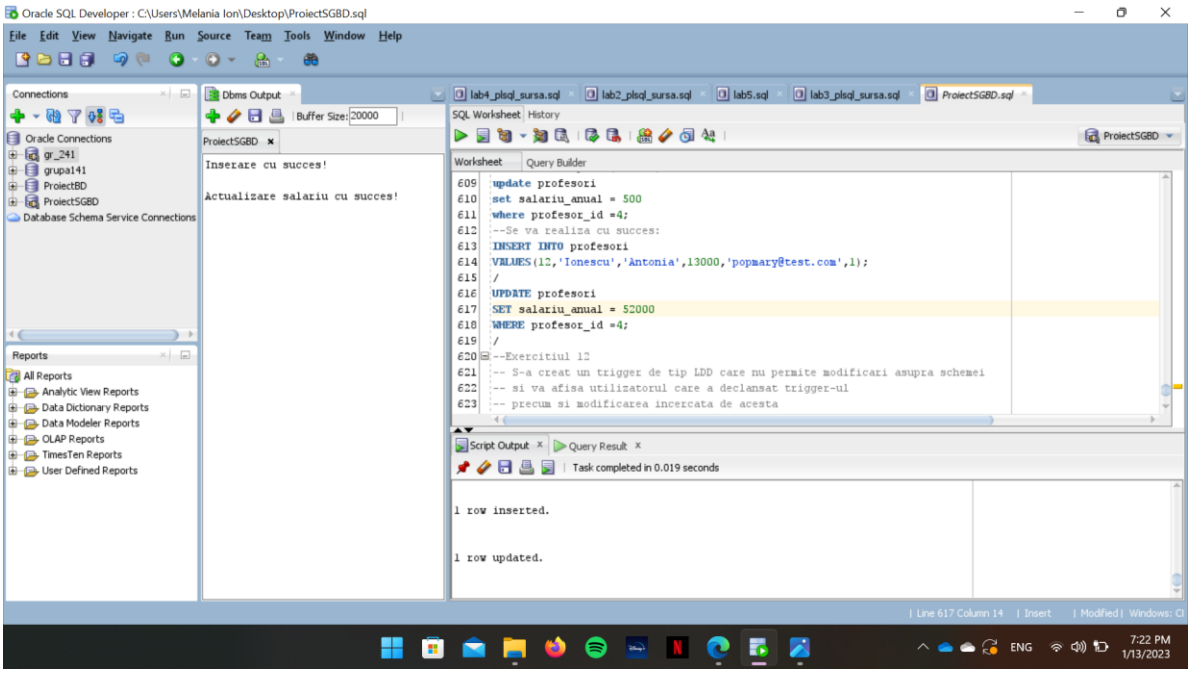
```
--DECLANSARE TRIGGER 11
--Incercare inserare (eroare)
INSERT INTO profesori       --salariu prea mare
VALUES(12,'Ionescu','Antonia',20000,'popmary@test.com',1);
INSERT INTO profesori       --salariu prea mic
VALUES(12,'Ionescu','Antonia',12000,'popmary@test.com',1);
--Incercare update(eroare)
update profesori
set salariu_anual = 500
where profesor_id =4;
--Se va realiza cu succes:
INSERT INTO profesori
VALUES(12,'Ionescu','Antonia',13000,'popmary@test.com',1);
/
UPDATE profesori
SET salariu_anual = 52000
WHERE profesor_id =4;
/
```




Oracle SQL Developer : C:\Users\Melania Ion\Desktop\ProiectSGBD.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
gr_241
grupa141
ProiectBD
ProiectSGBD
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Dbms Output
Buffer Size: 20000
ProiectSGBD
lab4_plsql_sursa.sql
lab2_plsql_sursa.sql
lab5.sql
lab3_plsql_sursa.sql
ProiectSGBD.sql
SQL Worksheet
History
Worksheet
Query Builder
ELSE
DBMS_OUTPUT.PUT_LINE('Actualizare salariu cu succes!');
END IF;
END IF;
END;
/
ALTER TRIGGER trig11 ENABLE;
ALTER TRIGGER trig11 DISABLE;
/
--Incercare inserare (eroare)
INSERT INTO profesori
VALUES(12,'Ionescu','Antonia',20000,'popmary@test.com',1);
--Incercare update(eroare)
update profesori
set salariu_anual = 500
Script Output
Query Result
Task completed in 0.038 seconds
VALUES(12,'Ionescu','Antonia',20000,'popmary@test.com',1)
Error report -
ORA-20002: Salariu este prea mare pentru un profesor nou!
ORA-06512: at "C##MELANIAION2.TRIG11", line 4
ORA-04088: error during execution of trigger 'C##MELANIAION2.TRIG11'
Line 604 Column 12
Insert
Modified
Windows: Cr
ENG
7:19 PM
1/13/2023


Oracle SQL Developer : C:\Users\Melania Ion\Desktop\ProiectSGBD.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
gr_241
grupa141
ProiectBD
ProiectSGBD
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Dbms Output
Buffer Size: 20000
ProiectSGBD
lab4_plsql_sursa.sql
lab2_plsql_sursa.sql
lab5.sql
lab3_plsql_sursa.sql
ProiectSGBD.sql
SQL Worksheet
History
Worksheet
Query Builder
ALTER TRIGGER trig11 ENABLE;
ALTER TRIGGER trig11 DISABLE;
/
--Incercare inserare (eroare)
INSERT INTO profesori --salariu prea mare
VALUES(12,'Ionescu','Antonia',20000,'popmary@test.com',1);
INSERT INTO profesori --salariu prea mic
VALUES(12,'Ionescu','Antonia',12000,'popmary@test.com',1);
--Incercare update(eroare)
update profesori
set salariu_anual = 500
where profesor_id =4;
--Se va realiza cu succes:
INSERT INTO profesori
VALUES(12,'Ionescu','Antonia',13000,'popmary@test.com',1);
Script Output
Query Result
Task completed in 0.02 seconds
VALUES(12,'Ionescu','Antonia',12000,'popmary@test.com',1)
Error report -
ORA-20007: Salariul este prea mic!
ORA-06512: at "C##MELANIAION2.TRIG11", line 7
ORA-04088: error during execution of trigger 'C##MELANIAION2.TRIG11'
Line 607 Column 41
Insert
Modified
Windows: Cr
ENG
7:20 PM
1/13/2023


Oracle SQL Developer : C:\Users\Melania Ion\Desktop\ProiectSGBD.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
gr_241
grupa141
ProiectBD
ProiectSGBD
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Dbms Output
Buffer Size: 20000
ProiectSGBD
lab4_plsql_sursa.sql
lab2_plsql_sursa.sql
lab5.sql
lab3_plsql_sursa.sql
ProiectSGBD.sql
SQL Worksheet History
Worksheet Query Builder
603 --Incercare inserare (eroare)
604 INSERT INTO profesori --salariu prea mare
605 VALUES(12,'Ionescu','Antonia',20000,'popmary@test.com',1);
606 INSERT INTO profesori --salariu prea mic
607 VALUES(12,'Ionescu','Antonia',12000,'popmary@test.com',1);
608 --Incercare update(eroare)
609 update profesori
610 set salariu_anual = 500
611 where profesor_id =4;
612 --Se va realiza cu succes:
613 INSERT INTO profesori
614 VALUES(12,'Ionescu','Antonia',13000,'popmary@test.com',1);
615 /
616 UPDATE profesori
617 SET salariu_anual = 52000
Script Output
Query Result
Task completed in 0.016 seconds
where profesor_id =4
Error report -
ORA-20003: Nu puteti micsora salariile!
ORA-06512: at "C##MELANIAION2.TRIG11", line 14
ORA-04088: error during execution of trigger 'C##MELANIAION2.TRIG11'
Line 611 Column 19 Insert Modified Windows: Cl
ENG
7:21 PM
1/13/2023


Oracle SQL Developer : C:\Users\Melania Ion\Desktop\ProiectSGBD.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
gr_241
grupa141
ProiectBD
ProiectSGBD
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Dbms Output
Buffer Size: 20000
ProiectSGBD
Inserare cu succes!
Actualizare salariu cu succes!
lab4_plsql_sursa.sql
lab2_plsql_sursa.sql
lab5.sql
lab3_plsql_sursa.sql
ProiectSGBD.sql
SQL Worksheet History
Worksheet Query Builder
609 update profesori
610 set salariu_anual = 500
611 where profesor_id =4;
612 --Se va realiza cu succes:
613 INSERT INTO profesori
614 VALUES(12,'Ionescu','Antonia',13000,'popmary@test.com',1);
615 /
616 UPDATE profesori
617 SET salariu_anual = 52000
618 WHERE profesor_id =4;
619 /
620 --Exercitiul 12
621 -- S-a creat un trigger de tip LDD care nu permite modificari asupra schemei
622 -- si va afisa utilizatorul care a declansat trigger-ul
623 -- precum si modificarea incercata de acesta
Script Output
Query Result
Task completed in 0.019 seconds
1 row inserted.
1 row updated.
Line 617 Column 14 Insert Modified Windows: Cl
ENG
7:22 PM
1/13/2023

Exercitiul 12: Definiți un trigger de tip LDD. Declanșați trigger-ul.

REZOLVARE: S-a creat un trigger de tip LDD care nu permite modificari asupra schemei, afisandu-se utilizatorul care a declansat trigger-ul, precum si modificarea incercata de acesta.

```
CREATE OR REPLACE TRIGGER trig12
AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
   DBMS_OUTPUT.PUT_LINE('Au fost incercate modificari de catre user-ul: ' || LOGIN_USER);
   DBMS_OUTPUT.PUT_LINE('Actiunea incercata de: -' || SYSEVENT || '- nu este permisa!' );
   RAISE_APPLICATION_ERROR(-20005,'Nu aveti dreptul sa faceti modificari!');
END;
--DECLANSARE TRIGGER 12
CREATE TABLE tabel_nou(
id number(10) NOT NULL, camp1 number(10), camp2 varchar2(50) NOT NULL, camp3 varchar2(50),
CONSTRAINT nou_pk PRIMARY KEY(id));
```