

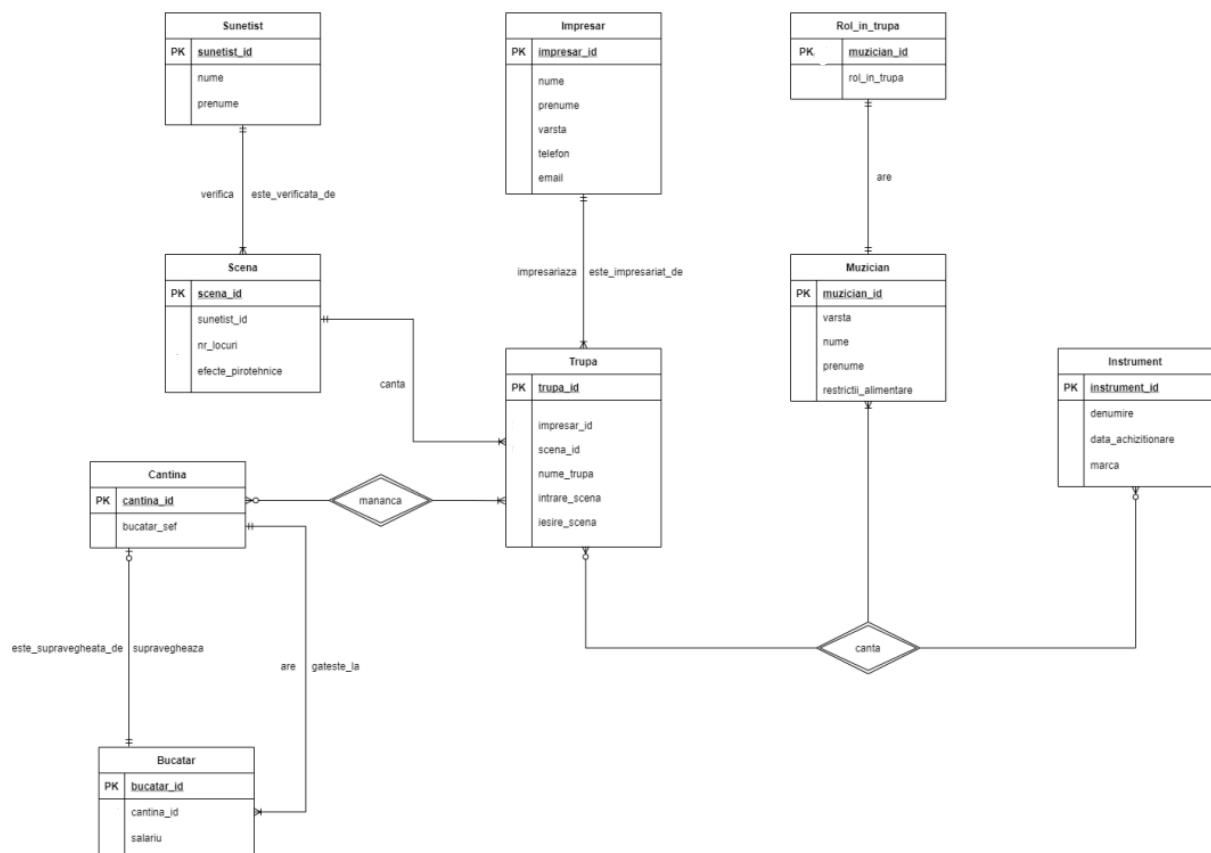
Bază de date pentru un festival de muzică

Introducere

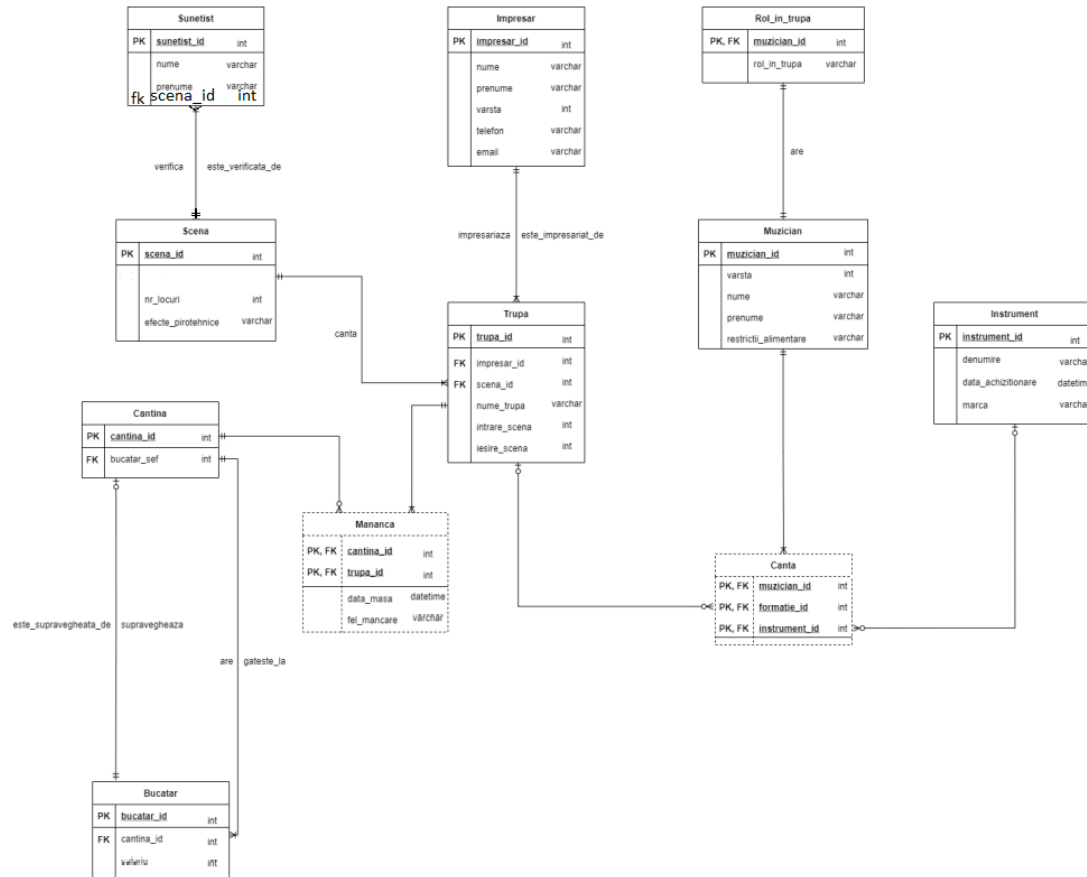
Obiectiv:

Organizatorii unui festival de muzică doresc implementarea unei baze de date care să îi ajute să țină evidența tuturor trupelor participante și a artiștilor. Unii artiști nu și-au adus de acasă instrumente, deoarece festivalul le poate pune la dispoziție și, de aceea, este nevoie să ținem evidența lor în baza de date. Pentru a putea lua legătura cu o trupă este necesar să fie contactat impresarul ei. De asemenea, artiștilor le este pusă la dispoziție zilnic de la 7:00 la 20:00 mâncare, în niște zone special amenajate (cantine) de către bucătari.

Cerința 2: diagrama entitate-relație(ERD)



Cerința 3: Diagrama conceptuală a modelului



Cerința 4 + 5: Implementarea în Oracle a diagramei conceptuală realizate: definirea tuturor tabelelor, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc) și popularea tabelor.

```
create table festival_muzician
(muzician_id int,
varsta int,
nume varchar(30),
prenume varchar(30),
restrictii_alimentare varchar(50),
constraint pk_muzician primary key (muzician_id));
```

```
create table festival_instrument
(instrument_id int,
denumire varchar(30) not null,
data_achizitionare date,
marca varchar(30),
constraint pk_instrument primary key (instrument_id));
```

Fig1. Crearea tabelului muzician

Fig2. Crearea tabelului instrument

```
create table festival_impresar
(impresar_id int,
nume varchar(30) not null,
prenume varchar(30),
varsta int,
telefon varchar(10) not null,
email varchar(20) not null,
constraint pk_impresar primary key (impresar_id));
```

Fig3. Crearea tabelului impresar

```
create table festival_scena
(scena_id int,
sunetist_id int not null,
nr_locuri int,
efecte_pirotehnice VARCHAR2(10),
constraint pk_scena primary key (scena_id));
```

Fig 4. Crearea tabelului scena

```
create table festival_sunetist
(sunetist_id int,
nume varchar(30),
prenume varchar(30),
scena_id int,
constraint pk_sunetist primary key (sunetist_id),
constraint fk_sunetist_scena foreign key (scena_id) references festival_scena(scena_id) on delete cascade);
```

Fig5. Crearea tabelului sunetist

```
create table festival_trupa
(trupa_id int,
impresar_id int not null,
scena_id int,
nume_trupa varchar(30),
intrare_scena int,
iesire_scena int,
constraint pk_trupa primary key (trupa_id),
constraint fk_trupa_impresar foreign key (impresar_id) references festival_impresar(impresar_id) on delete cascade,
constraint fk_trupa_scena foreign key (scena_id) references festival_scena(scena_id) on delete set null);
```

Fig6. Crearea tabelului trupa

```
create table festival_canta
(
muzician_id int,
formatie_id int,
instrument_id int,
constraint pk_canta primary key (muzician_id, formatie_id, instrument_id),
constraint fk_canta_muzician foreign key (muzician_id) references festival_muzician (muzician_id) on delete cascade,
constraint fk_canta_formatie foreign key (formatie_id) references festival_trupa (trupa_id) on delete cascade,
constraint fk_canta_instrument foreign key (instrument_id) references festival_instrument (instrument_id) on delete cascade);
```

Fig7. Crearea tabelului canta

```
create table festival_cantina
(
cantina_id int,
bucatar_sef int,
constraint pk_cantina primary key (cantina_id));
```

Fig8. Crearea tabelului cantina

```
create table festival_bucatar
(
bucatar_id int,
cantina_id int,
salariu int,
constraint pk_bucatar primary key (bucatar_id),
constraint fk_bucatar_cantina foreign key (cantina_id) references festival_cantina(cantina_id) on delete set null);
```

Fig9. Crearea tabelului bucătar

```
create table festival_mananca
(
cantina_id int,
trupa_id int,
data_masa date,
fel_mancare varchar(50),
constraint pk_mananca primary key (cantina_id, trupa_id),
constraint fk_mananca_cantina foreign key (cantina_id) references festival_cantina(cantina_id) on delete cascade,
constraint fk_mananca_trupa foreign key (trupa_id) references festival_trupa(trupa_id) on delete cascade);
```

Fig10. Crearea tabelului mănâncă

```
create table festival_rol_in_trupa
(
muzician_id int,
rol_in_trupa varchar(20),
constraint pk_rol_in_trupa primary key (muzician_id),
constraint fk_rol_in_trupa_muzician foreign key (muzician_id) references festival_muzician (muzician_id) on delete cascade);
```

Fig11. Crearea tabelului rol_in_trupa

Cerința 5: Adăugarea de date coerente în tabelele create

```
insert into festival_muzician
values(2, 40, 'Burlacianu', 'Bobo', null);

insert into festival_muzician
values(3, 12, 'Ciurescu', 'Andreea', null);

insert into festival_muzician
values(4, 2, 'Rotaru', 'Codrut', 'vegan');

insert into festival_muzician
values(5, 17, 'Suto', 'Bobi', 'pescetarian');
```

Fig12. Inserarea în tabela muzician

```
insert into festival_instrument
values (1, 'chitara',sysdate, 'Fender');

insert into festival_instrument
values (2, 'bass', sysdate - 1, 'Vero');

insert into festival_instrument
values (3, 'tobe', sysdate - 2, 'Marshall');

insert into festival_instrument
values (4, 'chitara', sysdate - 5, 'Fender');

insert into festival_instrument
values (5, 'pian', sysdate, 'Fender');
```

Fig13. Inserarea în tabela instrument

```
insert into festival_impresar
values(1, 'Parnescu', 'Adrian', 23, '0741123334', 'tamare1@gmail.com');

insert into festival_impresar
values(2, 'Tudose', 'Andrei', 21, '0711111111', 'tudose@gmail.com');

insert into festival_impresar
values(3, 'Ilie', 'Ion', 25, '0712345678', 'ion@gmail.com');

insert into festival_impresar
values(4, 'Lupu', 'Mihai', 22, '0751111222', 'mihailupu2@gmail.com');

insert into festival_impresar
values(5, 'Imre', 'Andrei', 20, '0751101001', 'imre@gmail.com');
```

Fig14. Inserarea în tabela impresar

```
insert into festival_sunetist
values (6, 'Ifri', 'Alexandru', 1);

insert into festival_sunetist
values (7, 'Blahovici', 'Radu', 2);

insert into festival_sunetist
values (8, 'Jitarasu', 'Catalina', 3);

insert into festival_sunetist
values (9, 'Epure', 'Codrin', 4);

insert into festival_sunetist
values (10, 'Agape', 'Dariana', 1);
```

Fig15. Inserarea în tabela sunetist

```
insert into festival_scena
values(6, 1, 50, 'true', 1);

insert into festival_scena
values(7, 5, 60, 'true', 2);

insert into festival_scena
values(8, 4, 30, 'false', 3);

insert into festival_scena
values(9, 1, 75, 'true', 4);

insert into festival_scena
values(10, 2, 100, 'false', 5);
```

Fig16. Inserarea în tabela scena

```
insert into festival_trupa
values (1, 1, 2, 'System of a Down', 18, 20);

insert into festival_trupa
values (2, 5, 2, 'Fara Zahar', 20, 21);

insert into festival_trupa
values (3, 5, 2, 'Suie Pararude', 22, 23);

insert into festival_trupa
values (4, 4, 4, 'Cargo', 17, 18);

insert into festival_trupa
values (5, 2, 1, 'Zob', 18, 19);
```

Fig17. Inserarea în tabela trupa

```

insert into festival_canta
values (1, 2, 2);

insert into festival_canta
values (2, 2, 1);

insert into festival_canta
values (3, 2, 5);

insert into festival_canta
values (4, 5, 1);

insert into festival_canta
values (5, 5, 3);

insert into festival_canta
values (2, 3, 2);

insert into festival_canta
values (4, 2, 1);

insert into festival_canta
values (5, 2, 3);

insert into festival_canta
values (4, 3, 2);

insert into festival_canta
values (1, 3, 1);

```

Fig18. Inserarea in tabela canta

```

insert into festival_cantina
values (1, null);

insert into festival_cantina
values (2, null);

insert into festival_bucatar
values (1, 1, 3000);

insert into festival_bucatar
values (2, 1, 3100);

update festival_cantina
set bucatar_sef = 1
where cantina_id = 1;

insert into festival_bucatar
values (3, 2, 4000);

insert into festival_bucatar
values (4, 2, 4000);

update festival_cantina
set bucatar_sef = 4
where cantina_id = 2;

insert into festival_cantina
values (3, null);

insert into festival_cantina
values (4, null);

insert into festival_cantina
values (5, null);

insert into festival_bucatar
values (6, 3, 3500);

insert into festival_bucatar
values (7, 5, 4050);

insert into festival_bucatar
values (8, 3, 4100);

insert into festival_bucatar
values (9, 4, 1000);

```

Fig19. Inserarea in tabelele bucatar si cantina

```

insert into festival_mananca
values(1, 1, sysdate, 'sushi');

insert into festival_mananca
values(2, 2, sysdate - 1, 'cartofi');

insert into festival_mananca
values(3, 1, sysdate - 1, 'paste');

insert into festival_mananca
values(4, 3, sysdate, 'pizza');

insert into festival_mananca
values(5, 4, sysdate, 'pizza');

insert into festival_mananca
values(1, 5, sysdate, 'clatite');

insert into festival_mananca
values(2, 1, sysdate, 'ciorba');

insert into festival_mananca
values(3, 2, sysdate, 'ciuperici');

```

Fig20. Inserarea în tabela mananca

```

insert into festival_rol_in_trupa
values (1, 'solist');

insert into festival_rol_in_trupa
values (2, 'chitarist');

insert into festival_rol_in_trupa
values (3, 'pianist');

insert into festival_rol_in_trupa
values (4, 'solist');

insert into festival_rol_in_trupa
values (5, 'tobosar');

```

Fig21. Inserarea în tabela rol_in_trupa

Cerința 6: Am creat o procedură prin care, pentru o scenă dată ca parametru, afișez ce sunetiști au verificat-o și ce trupe urmează să cânte pe ea

```
create or replace procedure cerinta6 (s_id festival_scena.scena_id%type)
```

```
as
```

```
--tablou de trupe
```

```
type tbl_idx is
```

```
    table of varchar2(30) index by pls_integer;
```

```
t_n tbl_idx;
```

```
--ce sunetisti
```

```
type vector is
```

```
    varray(20) of festival_sunetist%rowtype;
```

```
t_s vector:= vector();
```

```
begin
```

```
    select nume_trupa
```

```
    bulk collect into t_n
```

```
    from festival_trupa f
```

```
    where f.scena_id = s_id;
```

```
    dbms_output.put_line('Pe scena ' || s_id || ' canta ' || t_n.count() || ' trupe:');
```

```
    for i in t_n.first..t_n.last loop
```

```
        dbms_output.put_line(t_n(i));
```

```
    end loop;
```

```
    dbms_output.put_line('Scena a fost verificata de sunetistii: ');
```

```
    select *
```

```
bulk collect into t_s
from festival_sunetist sun
where sun.scena_id = s_id;

for i in t_s.first..t_s.last loop
    dbms_output.put_line(t_s(i).nume || ' ' || t_s(i).prenume);
end loop;
```

```
end;
```

```
/
```

```
begin
```

```
cerinta6('1');
```

```
end;
```

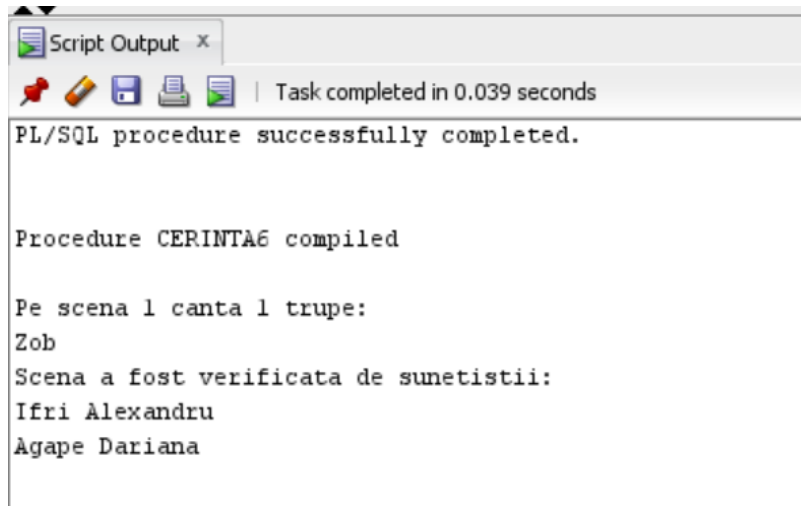
```
/
```

```
where sun.scena_id = s_id;

for i in t_s.first..t_s.last loop
    dbms_output.put_line(t_s(i).nume || ' ' || t_s(i).prenume);
end loop;

end;
/

begin
    cerinta6('1');
end;
/
```

Cerința 7: Pentru fiecare scenă afișez câte trupe cântă pe ea și salvez într-un cursor id-ul scenei, precum și numărul trupelor.

create or replace procedure cerinta7

as

 v_id_scena festival_scena.scena_id%type;

 v_ct number(3);

 cursor c is

 select s.scena_id, count(trupa_id)

 from festival_scena s join festival_trupa t on(s.scena_id = t.scena_id)

 group by s.scena_id;

begin

 open c;

 loop

 fetch c into v_id_scena, v_ct;

 exit when c%notfound;

 if v_ct = 0 then

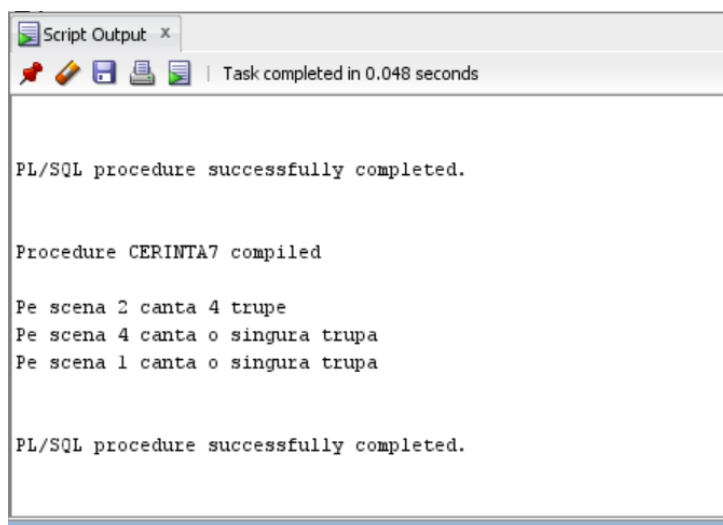
 dbms_output.put_line('Pe scena ' || v_id_scena || ' nu canta nicio trupa');

 elsif v_ct = 1 then

 dbms_output.put_line('Pe scena ' || v_id_scena || ' canta o singura trupa');

```
else
    --dmbs_output.put_line('Pe scena ' || v_id_scena || ' canta ' || v_ct || 'trupe');
    dbms_output.put_line('Pe scena ' || v_id_scena || ' canta ' || v_ct || ' trupe');
end if;
end loop;

close c;
end;
/
begin
    cerinta7();
end;
/
```



Cerința 8: --pentru un instrument al cărui nume este dat ca paramentru, câți muzicieni cântă la el, indiferent de trupa din care fac parte. Cele 3 tabele utilizate: festival_muzician, festival_canta, festival_instrument

Excepții: -no_data_found: nu există instrumentul dat ca parametru

-too_many_rows: un muzician cântă la mai multe instrumente

-alte excepții care pot apărea

create or replace function cerinta8(instr festival_instrument.denumire%type) return number

is

nr_muz number;

type tbl_idx is table of festival_instrument%rowtype index by pls_integer;

aux tbl_idx;

no_data_found exception;

too_many_rows exception;

begin

select * bulk collect into aux

from festival_instrument

where festival_instrument.denumire = instr;

if sql%notfound then

raise no_data_found;

end if;

select count(m.muzician_id) into nr_muz

from festival_muzician m join festival_canta c on (m.muzician_id = c.muzician_id)

join festival_instrument i on (c.instrument_id = i.instrument_id)

where i.denumire = instr;

return nr_muz;

exception

when no_data_found then

dbms_output.put_line('Nu canta nimeni la ' || instr);

return -1;

when too_many_rows then

dbms_output.put_line('Un muzician canta la mai multe instrumente');

return -1;

```
when others then

    dbms_output.put_line('Codul erorii: ' || sqlcode);

    dbms_output.put_line('Mesajul erorii: ' || sqlerrm);

    return -1;

end;

/

declare

    aux festival_instrument.denumire%type;

begin

    aux := cerinta8('chitara');

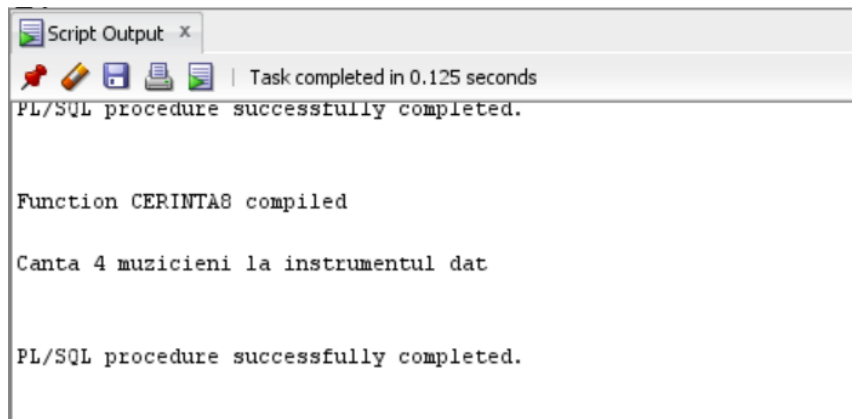
    if aux > -1 then

        dbms_output.put_line('Canta ' || aux || ' muzicieni la instrumentul dat');

    end if;

end;

/
```



Cerința 9: Vreau să afișez instrumentele la care cântă muzicienii care mănâncă la o cantină dată ca parametru. Cele 5 tabele utilizate: festival_cantina, festival_mananca, festival_trupa, festival_canta, festival_instrument.

Excepții: -too many rows un muzician care canta la mai multe instrumente

-no data found nu exista instrumentul

-alte excepții

create or replace procedure cerinta9(cantina festival_cantina.cantina_id%type)

as

type tbl_idx is table of festival_instrument.denumire%type index by pls_integer;
instrumente tbl_idx;

no_data_fount exception;

too_many_rows exception;

begin

select i.denumire bulk collect into instrumente
from festival_cantina c join festival_mananca m on (c.cantina_id = m.cantina_id)
join festival_trupa t on (m.trupa_id = t.trupa_id)
join festival_canta canta on (t.trupa_id = canta.formatie_id)
join festival_instrument i on (canta.instrument_id = i.instrument_id)
where c.cantina_id = cantina;

for i in instrumente.first..instrumente.last loop
 dbms_output.put_line(instrumente(i));
end loop;

exception

when no_data_found then
 dbms_output.put_line('Nu exista cantina ' || cantina);

when too_many_rows then
 dbms_output.put_line('Unul dintre muzicieni canta la mai multe instrumente');

```
end;
```

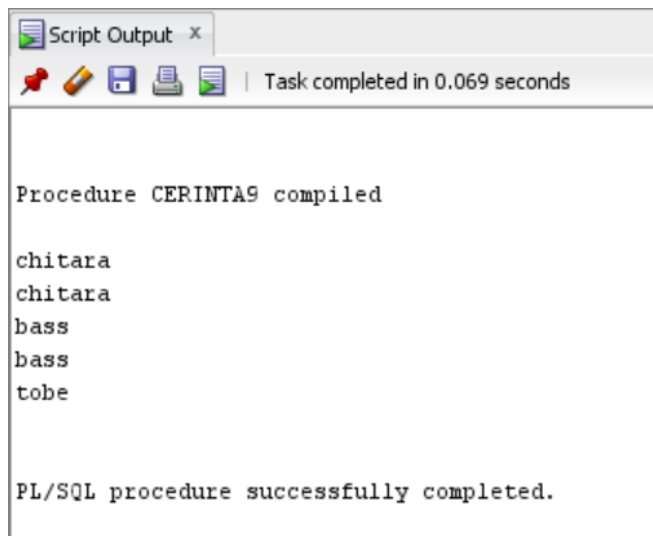
```
/
```

```
begin
```

```
    cerinta9(1);
```

```
end;
```

```
/
```



Cerința 10: Am făcut un trigger care nu permite actualizarea tabeli trupa într-o zi nelucrătoare(25 decembrie sau 1 ianuarie)

****am adăugat si data curentă pentru a exemplifica***

```
create or replace trigger cerinta10
```

```
    before insert or update or delete on festival_trupa
```

```
begin
```

```
    if(to_char(sysdate, 'DD/MM') = '25/12' or to_char(sysdate, 'DD/MM')= '01/01' or
```

```
    to_char(sysdate, 'DD/MM') = '07/01') then
```

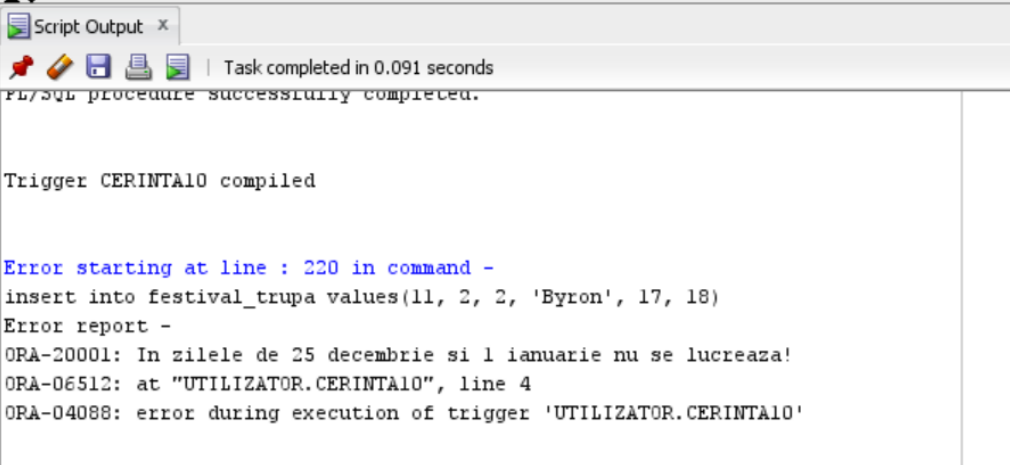
```
        raise_application_error(-20001, 'In zilele de 25 decembrie si 1 ianuarie nu se lucreaza!');
```

```
    end if;
```

```
end;
```

/

```
insert into festival_trupa values(11, 2, 2, 'Byron', 17, 18);
```



The screenshot shows a 'Script Output' window with a toolbar at the top. The status bar indicates 'Task completed in 0.091 seconds'. The main text area displays the following output:

```
PL/SQL procedure successfully completed.

Trigger CERINTA10 compiled

Error starting at line : 220 in command -
insert into festival_trupa values(11, 2, 2, 'Byron', 17, 18)
Error report -
ORA-20001: In zilele de 25 decembrie si 1 ianuarie nu se lucreaza!
ORA-06512: at "UTILIZATOR.CERINTA10", line 4
ORA-04088: error during execution of trigger 'UTILIZATOR.CERINTA10'
```

Cerința 11: Am realizat un declanșator care să nu permită micșorarea vârstei muzicienilor

```
create or replace trigger trig2
```

```
    before update of varsta on festival_muzician
```

```
    for each row
```

```
begin
```

```
    if(:new.varsta < :old.varsta) then
```

```
        raise_application_error(-20002, 'varsta nu poate fi micsorata!');
```

```
    end if;
```

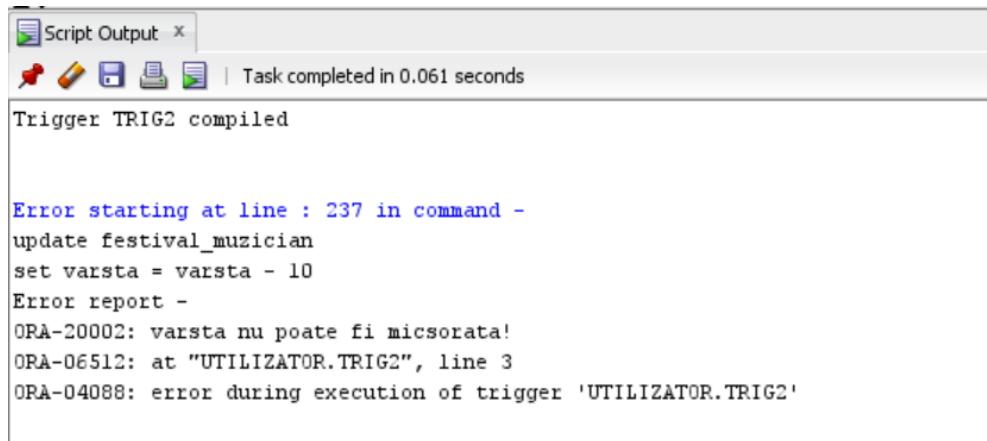
```
end;
```

/

```
update festival_muzician
```

```
set varsta = varsta - 10;
```

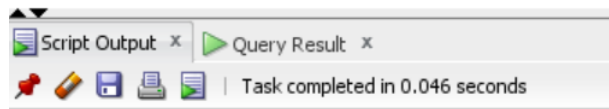
```
drop trigger trig2;
```



Cerința 12: Am creat un trigger care se declanșează de fiecare dată când execut o operație de tip LDD, inserând în tabela nou creată un log cu detaliile operației

```
create table audit_user(  
    user_logat VARCHAR2(30),  
    nume_bd VARCHAR2(50),  
    eveniment VARCHAR2(20),  
    nume_obiect_referit VARCHAR2(30),  
    data DATE);  
  
create or replace trigger trig3  
    after create or drop or alter on schema  
  
begin  
    insert into audit_user values (sys.login_user, sys.database_name, sys.sysevent,  
sys.dictionar_obj_name, sysdate);  
  
end;  
  
/  
  
create table tabel_aux(col1_aux number(2));  
alter table tabel_aux add(col2_aux number(2));  
create index ind on tabel_aux(col1_aux);  
drop table tabel_aux;  
select * from audit_user;
```


drop trigger trig3;



Trigger TRIG3 compiled

Table TABEL_AUX created.

Table TABEL_AUX altered.

Index IND created.

Table TABEL_AUX dropped.

A screenshot of a database management tool's 'Query Result' window. The window shows a table with 5 columns: USER_LOGAT, NUME_BD, EVENIMENT, NUME_OBJECT_REFERIT, and DATA. The table contains 12 rows of data, all dated 07-JAN-22. The status bar indicates 'All Rows Fetched: 12 in 0.002 seconds'.

	USER_LOGAT	NUME_BD	EVENIMENT	NUME_OBJECT_REFERIT	DATA
1	UTILIZATOR	XE	CREATE	TABEL_AUX	07-JAN-22
2	UTILIZATOR	XE	ALTER	TABEL_AUX	07-JAN-22
3	UTILIZATOR	XE	CREATE	IND	07-JAN-22
4	UTILIZATOR	XE	DROP	TABEL_AUX	07-JAN-22
5	UTILIZATOR	XE	CREATE	TABEL_AUX	07-JAN-22
6	UTILIZATOR	XE	ALTER	TABEL_AUX	07-JAN-22
7	UTILIZATOR	XE	CREATE	IND	07-JAN-22
8	UTILIZATOR	XE	DROP	TABEL_AUX	07-JAN-22
9	UTILIZATOR	XE	CREATE	TABEL_AUX	07-JAN-22
10	UTILIZATOR	XE	ALTER	TABEL_AUX	07-JAN-22
11	UTILIZATOR	XE	CREATE	IND	07-JAN-22
12	UTILIZATOR	XE	DROP	TABEL_AUX	07-JAN-22

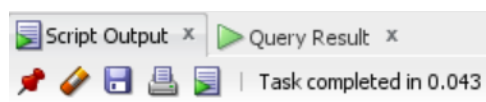


Table TABEL_AUX dropped.

Trigger TRIG3 dropped.



Cerința 13:

create or replace package proiect

as

 procedure cerinta6 (s_id festival_scena.scena_id%type);

 procedure cerinta7;

 function cerinta8(instr festival_instrument.denumire%type) return number;

 procedure cerinta9(cantina festival_cantina.cantina_id%type);

end proiect;

/

create or replace package body proiect

as

 procedure cerinta6 (s_id festival_scena.scena_id%type)

 as

 --tablou de trupe

 type tbl_idx is

 table of varchar2(30) index by pls_integer;

 t_n tbl_idx;

 --sunetisti

 type vector is

 varray(20) of festival_sunetist%rowtype;

 t_s vector:= vector();

begin

 select nume_trupa

```
bulk collect into t_n
from festival_trupa f
where f.scena_id = s_id;

dbms_output.put_line('Pe scena ' || s_id || ' canta ' || t_n.count() || ' trupe:');

for i in t_n.first..t_n.last loop
    dbms_output.put_line(t_n(i));
end loop;

dbms_output.put_line('Scena a fost verificata de sunetistii: ');

select *
bulk collect into t_s
from festival_sunetist sun
where sun.scena_id = s_id;

for i in t_s.first..t_s.last loop
    dbms_output.put_line(t_s(i).nume || ' ' || t_s(i).prenume);
end loop;

end;

procedure cerinta7
as
    v_id_scena festival_scena.scena_id%type;
    v_ct number(3);
    cursor c is
        select s.scena_id, count(trupa_id)
        from festival_scena s join festival_trupa t on(s.scena_id = t.scena_id)
        group by s.scena_id;
begin
```

```
open c;

loop

    fetch c into v_id_scena, v_ct;

    exit when c%notfound;

    if v_ct = 0 then

        dbms_output.put_line('Pe scena ' || v_id_scena || ' nu canta nicio trupa');

    elsif v_ct = 1 then

        dbms_output.put_line('Pe scena ' || v_id_scena || ' canta o singura trupa');

    else

        --dbms_output.put_line('Pe scena ' || v_id_scena || ' canta ' || v_ct || ' trupe');

        dbms_output.put_line('Pe scena ' || v_id_scena || ' canta ' || v_ct || ' trupe');

    end if;

end loop;

close c;

end;

function cerinta8(instr festival_instrument.denumire%type) return number
is
    nr_muz number;
    type tbl_idx is table of festival_instrument%rowtype index by pls_integer;
    aux tbl_idx;
    no_data_fount exception;

begin
    select * bulk collect into aux
    from festival_instrument
    where festival_instrument.denumire = instr;

    if sql%notfound then
```

```
        raise no_data_found;
    end if;

    select count(m.muzician_id) into nr_muz
    from festival_muzician m join festival_canta c on (m.muzician_id = c.muzician_id)
    join festival_instrument i on (c.instrument_id = i.instrument_id)
    where i.denumire = instr;

    return nr_muz;

exception
    when no_data_found then
        dbms_output.put_line('Nu canta nimeni la ' || instr);
        return -1;
    when others then
        dbms_output.put_line('Codul erorii: ' || sqlcode);
        dbms_output.put_line('Mesajul erorii: ' || sqlerrm);
        return -1;
end;

procedure cerinta9(cantina festival_cantina.cantina_id%type)
as

    type tbl_idx is table of festival_instrument.denumire%type index by pls_integer;
    instrumente tbl_idx;

    no_data_fount exception;
    too_many_rows exception;

begin
```

```
select i.denumire bulk collect into instrumente
from festival_cantina c join festival_mananca m on (c.cantina_id = m.cantina_id)
join festival_trupa t on (m.trupa_id = t.trupa_id)
join festival_canta canta on(t.trupa_id = canta.formatie_id)
join festival_instrument i on(canta.instrument_id = i.instrument_id)
where c.cantina_id = cantina;

for i in instrumente.first..instrumente.last loop
    dbms_output.put_line(instrumente(i));
end loop;

exception
when no_data_found then
    dbms_output.put_line('Nu exista cantina ' || cantina);

when too_many_rows then
    dbms_output.put_line('Unul dintre muzicieni canta la mai multe instrumente');

end;
end proiect;
/
```

```
create or replace package body proiect
as
  procedure cerinta6 (s_id festival_scena.scena_id%type)
  as
    --tablou de trupe
    type tbl_idx is
      table of varchar2(30) index by pls_integer;
    t_n tbl_idx;

    --sunetisti
    type vector is
      varray(20) of festival_sunetist%rowtype;
    t_s vector:= vector();
```

Script Output x Query Result x

Task completed in 0.046 seconds

Package Body PROIECT compiled

Cerința 14: Am creat un pachet în care memorăm informații pe care deja le știm despre muzicieni într-o colecție

-tipuri de date complexe: înregistrări si tabel indexat de înregistrări

-funcții: -verificăm pentru fiecare muzician dacă e major

-verificăm dacă are restricții alimentare

-proceduri: -creare

-afișare

create or replace package cerinta14

as

```
type rec is record (
  muzician_id number,
  nume varchar2(50),
  varsta number,
  restrictii_alimentare varchar2(50)
);
```

muz rec;

type tbl is table of rec index by pls_integer;

muzicieni tbl := tbl();

```
procedure creare;  
function muzician_major(x number) return boolean;  
function restrictii_alimentare(x number) return festival_muzician.restrictii_alimentare%type;  
procedure afisare;  
  
end cerinta14;  
/  
  
create or replace package body cerinta14  
as  
    procedure creare  
    is  
    begin  
  
        select muzician_id, nume, varsta, restrictii_alimentare  
        bulk collect into muzicieni  
        from festival_muzician;  
  
    end creare;  
    function muzician_major(x number) return boolean  
    is  
        varsta number;  
    begin  
        select m.varsta into varsta  
        from festival_muzician m  
        where m.muzician_id = x;  
        if varsta >= 18 then  
            return true;  
        else
```



```
        return false;
    end if;
end muzician_major;

function restrictii_alimentare(x number) return festival_muzician.restrictii_alimentare%type
is
    restrictie varchar2(50);
begin
    select m.restrictii_alimentare
    into restrictie
    from festival_muzician m
    where m.muzician_id = x;
    return restrictie;
end restrictii_alimentare;
```

```
procedure afisare
is
begin
    for i in muzicieni.first..muzicieni.last loop
        muz := muzicieni(i);
        if muzician_major(muz.muzician_id) then
            dbms_output.put_line('Muzicianul ' || muz.numa || ' este major si ');
            if restrictii_alimentare(muz.muzician_id) is not null then
                dbms_output.put_line('are restrictia alimentara: ' || muz.restrictii_alimentare);
            else
                dbms_output.put_line('nu are restrictii alimentare');
            end if;
        else
            dbms_output.put_line('Muzicianul ' || muz.numa || ' este MINOR si ');
            if restrictii_alimentare(muz.muzician_id) is not null then
                dbms_output.put_line('are restrictia alimentara: ' || muz.restrictii_alimentare);
            else
```

```
        dbms_output.put_line('nu are restrictii alimentare');
    end if;
end if;
end loop;
end afisare;

end cerinta14;

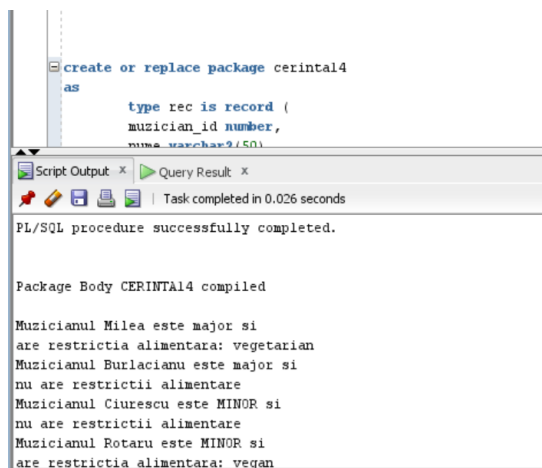
/

begin
    cerinta14.creare();
    cerinta14.afisare();
end;

/

rollback;

/
```



```
create or replace package cerinta14
as
    type rec is record (
        muzician_id number,
        nume varchar2(50)
    );
    procedure creare;
    procedure afisare;
end package cerinta14;

/

begin
    cerinta14.creare();
    cerinta14.afisare();
end;

/

rollback;

/
```

Script Output x Query Result x

Task completed in 0.026 seconds

PL/SQL procedure successfully completed.

Package Body CERINTA14 compiled

Muzicianul Milea este major si
are restrictia alimentara: vegetarian
Muzicianul Burlacianu este major si
nu are restrictii alimentare
Muzicianul Ciurescu este MINOR si
nu are restrictii alimentare
Muzicianul Rotaru este MINOR si
are restrictia alimentara: vegan

