

SUBIECTE ATESTAT BAZE DE DATE – PROBLEME PROPUSE
An școlar 2023 – 2024

1. Să se construiască o bază de date care să conțină tabela **CONCURS** având câmpurile: *cod, nume, rasa, proprietar, varsta, premiu* (I, II, III, M sau necompletat). Tabela conține datele unor câini participanți la un concurs.

a) Să se adauge în tabelă 5 înregistrări.

b) Să se afișeze datele câștilor care au obținut premiul I, în ordine alfabetică după nume.

c) Să se afișeze datele participanților care au vârsta mai mare decât numărul premiilor III acordate.

```
create table concurs(cod number(3), nume varchar2(30), rasa varchar2(30), proprietar varchar2(50), varsta number(2), premiu varchar2(3))
```

```
insert into concurs values (1, 'Bubi', 'lup', 'Ionescu', 15, 'I')
```

```
insert into concurs values (2, 'Bobi', 'lup', 'Popescu', 16, 'I')
```

```
insert into concurs values (3, 'Bubu', 'lup', 'Stanescu', 1, 'III')
```

```
insert into concurs values (4, 'Bibi', 'lup', 'Vasilescu', 2, 'III')
```

```
insert into concurs values (5, 'Bobo', 'lup', 'Georgescu', 10, NULL)
```

```
select *  
from concurs  
where premiu= 'I'  
order by nume
```

```
select *  
from concurs  
where varsta > (select count(*)  
                from concurs  
                where premiu= 'III')
```

2. Să se construiască o bază de date care să conțină tabela **CONCURS** având câmpurile: *cod, nume, rasa, proprietar, varsta, premiu* (I, II, III, M sau necompletat). Tabela conține datele unor câini participanți la un concurs.

a) Să se adauge în tabelă 5 înregistrări.

b) Să se afișeze datele participanților care nu au obținut niciun premiu, în ordine descrescătoare după vârstă.

c) Să se afișeze datele participanților care au vârsta mai mică decât cel mai tânăr câștigător al premiului I.

```
select *  
from concurs  
where premiu is NULL  
order by varsta desc
```

```
select*  
from concurs  
where varsta < (select min(varsta)  
                from concurs  
                where premiu = 'I')
```

3. Să se construiască o bază de date care să conțină tabela **CONCURS** având câmpurile: *cod, nume, rasa, proprietar, varsta, premiu* (I, II, III, M sau necompletat), *sectiune*. Tabela conține datele unor câini participanți la un concurs.

a) Să se adauge în tabelă 5 înregistrări.

b) Să se afișeze datele participanților din rasa Beagle care au obținut mențiune.

c) Să se afișeze datele participanților care au aceeași rasă ca și cățelul Aris. Se presupune că există un singur participant cu numele Aris.

```
create table concurs(cod number(3), nume varchar2(30), rasa varchar2(30), proprietar varchar2(50), varsta number(2), premiu varchar2(3), sectiune varchar2(30))
```

```
insert into concurs values (1, 'Aris', 'lup', 'Ionescu', 15, 'I', 'defilare')
insert into concurs values (2, 'Bobi', 'lup', 'Vasilescu', 16, 'I', 'defilare')
insert into concurs values (3, 'Bubu', 'Beagle', 'Stanescu', 1, 'III', 'defilare')
insert into concurs values (4, 'Rudy', 'Bichon', 'Vasilescu', 2, 'III', 'defilare')
insert into concurs values (5, 'Bobo', 'Beagle', 'Vasilescu', 10, 'M', 'defilare')
```

```
select *
from concurs
where rasa='Beagle' and premiu='M'
```

```
select *
from concurs
where rasa = (select rasa
              from concurs
              where nume = 'Aris')
and nume!='Aris'
```

4. Să se construiască o bază de date care să conțină tabela **CONCURS** având câmpurile: *cod*, *nume*, *rasa*, *proprietar*, *varsta*, *premiu* (I, II, III, M sau necompletat), *sectiune*. Tabela conține datele unor câini participanți la un concurs.

a) Să se adauge în tabelă 5 înregistrări.

b) Să se afișeze datele participanților cu vârsta mai mică de 5 ani, în ordine crescătoare după cod.

c) Să se afișeze datele participanților la aceeași secțiune ca și cățelul Aris. Se presupune că există un singur participant cu numele Aris.

```
select *
from concurs
where varsta<5
order by cod
```

```
select *
from concurs
where sectiune=(select sectiune
                from concurs
                where nume='Aris')
and nume!='Aris'
```

5. Să se construiască o bază de date care să conțină tabela **CONCURS** având câmpurile: *cod*, *nume*, *rasa*, *proprietar*, *varsta*, *premiu* (I, II, III, M sau necompletat). Tabela conține datele unor câini participanți la un concurs.

a) Să se adauge în tabelă 5 înregistrări.

b) Să se afișeze datele participanților din rasele Beagle, Labrador și Bichon.

c) Să se afișeze datele tuturor participanților care au aceeași vârstă ca și cățelul Aris. Se presupune că există un singur participant cu numele Aris.

```
select *
from concurs
where rasa in ('Beagle', 'Labrador', 'Bichon')
```

```
select *
from concurs
```

```
where varsta=(select varsta
                from concurs
                where nume='Aris')
and nume!='Aris'
```

6. Să se construiască o bază de date care să conțină tabela **CONCURS** având câmpurile: *cod*, *nume*, *rasa*, *proprietar*, *varsta*, *premiu* (I, II, III, M sau necompletat). Tabela conține datele unor câini participanți la un concurs.

- Să se adauge în tabelă 5 înregistrări.
- Să se afișeze datele participanților care au obținut premiul I și au peste 8 ani.
- Să se afișeze numele, rasa și vârsta participanților care au același proprietar ca și cățelul Rudy din rasa Bichon.

```
select *
from CONCURS
where varsta>8 and premiu='I'

select nume, rasa, varsta
from CONCURS
where proprietar=(select proprietar
                  from CONCURS
                  where nume='Rudy' and rasa='Bichon')
and nume!='Rudy'
```

7. Să se construiască o bază de date care să conțină tabela **CONCURS** având câmpurile: *cod*, *nume*, *rasa*, *proprietar*, *varsta*, *premiu* (I, II, III, M sau necompletat). Tabela conține datele unor câini participanți la un concurs.

- Să se adauge în tabelă 5 înregistrări.
- Să se afișeze media vârstei participanților care au obținut premiul II.
- Să se afișeze datele participanților care au obținut același premiu ca și candidatul cu codul citit de la tastatură.

```
select avg(varsta)
from concurs
where premiu='II'

SELECT *
FROM CONCURS
WHERE PREMIU=(SELECT PREMIU
               FROM CONCURS
               WHERE cod=:X)
```

8. Să se construiască o bază de date care să conțină tabela **CONCURS** având câmpurile: *cod*, *nume*, *rasa*, *proprietar*, *varsta*, *premiu* (I, II, III, M sau necompletat), *sectiune*. Tabela conține datele unor câini participanți la un concurs.

- Să se adauge în tabelă 5 înregistrări.
- Să se afișeze câte premii I au obținut participanții dintr-o rasă citită de la tastatură.
- Să se afișeze datele participanților care au aceeași vârstă ca și participantul cu codul 1234.

```
select count(premii)
from Concurs
where premiu='I' and rasa=:x

select nume, rasa, proprietar
from concurs
where varta=(select varsta
```

```
from concus
where cod=1234)
and cod!=1234
```

9. Să se construiască o bază de date care să conțină tabela **CONCURS** având câmpurile: *cod, nume, rasa, proprietar, varsta, premiu* (I, II, III, M sau necompletat). Tabela conține datele unor câini participanți la un concurs.

a) Să se adauge în tabelă 5 înregistrări.

b) Să se afișeze datele participanților la concurs care aparțin unei persoane cu numele citit de la tastatură.

c) Să se afișeze datele celor mai tineri participanți.

```
select *
from concurs
where nume = :A
```

```
select *
from concurs
where varsta = (select min(varsta)
                from concurs)
```

10. Să se construiască o bază de date care să conțină tabela **CONCURS** având câmpurile: *cod, nume, rasa, proprietar, varsta, premiu* (I, II, III, M sau necompletat), *sectiune*. Tabela conține datele unor câini participanți la un concurs.

a) Să se adauge în tabelă 5 înregistrări.

b) Să se afișeze câte premii a obținut un participant cu codul citit de la tastatura, știind că un câine poate participa la mai multe secțiuni ale concursului.

c) Să se afișeze datele participanților la concurs care au cea mai mare vârstă.

```
select count(premiu)
from concurs
where cod=:X
```

```
select *
from concurs
where varsta=(select max(varsta)
               from concurs)
```

11. Evidența cărților dintr-o bibliotecă este realizată cu ajutorul unei baze de date, care conține tabela **CARTI**, cu structura: *cod, titlu, autor, an_aparitie*.

a) Să se adauge în tabelă 5 înregistrări.

b) Să se afișeze numărul cărților din bibliotecă, apărute în perioada 2010-2020;

c) Să se afișeze lista autorilor care au în bibliotecă același număr de cărți ca și autorul x, x fiind citit de la tastatură.

```
SELECT count(cod)
FROM carti
WHERE an_aparitie between 2010 and 2020
```

```
select autor, count(titlu)
from carti
group by autor
having count(titlu)=(select count(titlu)
                     from carti
                     where autor=:x)
```

and autor!=:x

12. Evidența cărților dintr-o bibliotecă este realizată cu ajutorul unei baze de date, care conține tabela **CARTI**, cu structura: *cod, titlu, autor, an_aparitie*.

- Să se adauge în tabelă 5 înregistrări.
- Să se afișeze numărul cărților ce aparțin autorului x, x fiind citit de la tastatură;
- Să se afișeze lista cărților (titlu și autor) care au apărut cel mai recent.

```
SELECT count(titlu)
FROM CARTI
WHERE lower(autor)=lower(:x)

SELECT titlu, autor
FROM CARTI
WHERE an_aparitie=(SELECT max(an_aparitie)
                    FROM CARTI)
```

13. Evidența cărților dintr-o bibliotecă este realizată cu ajutorul unei baze de date, care conține tabela **CARTI**, cu structura: *cod, titlu, autor, nr_exemplare*.

- Să se adauge în tabelă 5 înregistrări.
- Să se afișeze lista cărților cu coduri formate din exact 3 cifre, ordonată alfabetic după titlu;
- Să se afișeze lista autorilor care au în bibliotecă cel mai mare număr total de exemplare.

```
SELECT *
FROM CARTI
WHERE LENGTH(cod)=3
ORDER BY titlu

SELECT autor
FROM CARTI
GROUP BY autor
HAVING SUM(nr_exemplare)=(SELECT MAX(SUM(nr_exemplare))
                          FROM CARTI
                          GROUP BY autor)
```

14. Evidența cărților dintr-o bibliotecă este realizată cu ajutorul unei baze de date, care conține tabela **CARTI**, cu structura: *cod, titlu, autor, nr_exemplare*.

- Să se adauge în tabelă 5 înregistrări.
- Să se afișeze numărul total de exemplare al cărților existente în bibliotecă;
- Să se afișeze lista autorilor care au cele mai multe titluri existente în bibliotecă.

```
create table carti(cod number(3), titlu varchar2(100), autor varchar2(100), nr_exemplare
number(4))

insert into carti values (1, 'Ion', 'Liviu Rebreanu', 25)
insert into carti values (2, 'Padurea spanzuratilor', 'Liviu Rebreanu', 42)
insert into carti values (3, 'Enigma Otiliei', 'George Calinescu', 15)
insert into carti values (4, 'Ultima noapte', 'Camil Petrescu', 121)
insert into carti values (5, 'Ciuleandra', 'Liviu Rebreanu', 23)

select sum(nr_exemplare) as "Nr. Total de exemplare"
from carti

select autor
from carti
group by autor
having count(autor)=(select max(count(autor))
                    from carti)
```

group by autor)

15. Evidența produselor dintr-un magazin este realizată cu ajutorul unei baze de date, care conține tabela **PRODUSE**, cu structura: *cod, denumire, cantitate, pret*. Pot exista produse cu aceeași denumire, care se vând cu prețuri diferite.

- Să se adauge în tabelă 5 înregistrări.
- Să se afișeze prețul mediu pentru fiecare produs, afișat cu două zecimale;
- Să se afișeze lista produselor vândute la preț minim.

```
select to_char(avg(pret), '9999.99') as "pret mediu"
from PRODUSE
group by denumire
```

```
select *
from PRODUSE
where pret=(select min(pret)
            from PRODUSE)
```

16. Evidența produselor dintr-un magazin este realizată cu ajutorul unei baze de date, care conține tabela **PRODUSE**, cu structura: *denumire, cantitate, furnizor, pret*. Pot exista produse cu aceeași denumire, care se vând cu prețuri diferite. Pot exista produse cu denumiri și prețuri identice, care au furnizori diferiți.

- Să se adauge în tabelă 5 înregistrări.
- Să se afișeze lista produselor, grupate după denumire și ordonate descrescător după cantitate;
- Să se afișeze lista produselor (cod, denumire) care există în cantitate minimă, ordonată crescător după codul produselor.

```
select *
from produse
order by denumire, cantitate desc
```

```
select cod, denumire
from produse
where pret=(select min(pret)
            from produse)
order by cod
```

17. Evidența produselor dintr-un magazin este realizată cu ajutorul unei baze de date, care conține tabela **PRODUSE**, cu structura: *cod, denumire, cantitate, pret*. Pot exista produse cu aceeași denumire, care sunt în cantități diferite și se vând cu prețuri diferite.

- Să se adauge în tabelă 5 înregistrări.
- Să se afișeze cantitatea totală existentă pentru produsul cu denumirea x, citită de la tastatură;
- Să se afișeze lista produselor (cod, denumire) care există în cantitate maximă, ordonată descrescător după denumirea produselor.

```
create table produse(cod number(3), denumire varchar2(30), cantitate number(3), pret number(7, 2))
insert into produse values(123, 'caiet', 100, 5)
insert into produse values(25, 'penar', 300, 5)
insert into produse values(325, 'caiet', 234, 10)
insert into produse values(46, 'penar', 200, 10)
insert into produse values(517, 'penar', 100, 5)
```

```
select sum(cantitate)
from produse
where denumire =:PRODUS
```

```

select cod, denumire
from produse
where cantitate=(select max(cantitate)
                  from produse)
order by denumire desc

```

18. Evidența notelor la informatică pentru elevii unei clase a 9-a este realizată cu ajutorul unei baze de date, care conține tabela **CATALOG**, cu structura: *nume, nota1, nota2, teza*. Dacă un elev nu a primit oricare din cele două note sau nu a dat teză, câmpurile respective conțin valoarea NULL.

a) Să se adauge în tabelă 5 înregistrări.

b) Să se afișeze numele elevilor care nu au dat teză sau nu au două note, în ordine alfabetică;

c) Să se afișeze numele elevilor pentru care nota1 este mai mare decât media notelor1 obținute de întreaga clasă.

```

create table catalog(nume varchar2(30), nota1 number(2), nota2 number(2), teza number(2))
insert into catalog values('Ionescu Dan', 10, null, 5)
insert into catalog values('Popescu Ana', null, null, null)
insert into catalog values('Stanescu Ion', 8, 7, null)
insert into catalog values('Vasilescu Anca', 10, 10, 10)
insert into catalog values('Georgescu Dana', 7, 8, 8)

```

```

select nume
from catalog
where teza is null or nota1 is null or nota2 is null
order by nume

```

```

select nume
from catalog
where nota1 > (select avg(nota1)
              from catalog)

```

19. Evidența mediilor generale pe semestrul I pentru elevii unei școli este realizată cu ajutorul unei baze de date, care conține tabela **CATALOG**, cu structura: *id, nume, clasa, media*.

a) Să se adauge în tabelă 5 înregistrări.

b) Să se afișeze media generală a fiecărei clase;

c) Să se afișeze numele elevilor de clasa a XII-a, care au medii în intervalul [9, 10].

```

create table catalog(id number(3), nume varchar2(30), clasa varchar2(4), media number(5, 2))
insert into catalog values(1, 'Ionescu Dan', 'XIIA', 10)
insert into catalog values(2, 'Popescu Ana', 'XIIB', 8.75)
insert into catalog values(3, 'Stanescu Ion', 'XIA', 8)
insert into catalog values(4, 'Vasilescu Anca', 'XIIA', 10)
insert into catalog values(5, 'Georgescu Dana', 'XIIB', 7.80)

```

```

select clasa, avg(media)
from catalog
group by clasa

```

```

select nume, clasa
from catalog
where lower(substr(clasa, 1, 3)) = 'xii' and media between 9 and 10

```

20. Evidența mediilor generale pe semestrul I pentru elevii unei școli este realizată cu ajutorul unei baze de date, care conține tabela **CATALOG**, cu structura: *id, nume, clasa, media*.

a) Să se adauge în tabelă 5 înregistrări.

b) Să se afișeze media maximă a fiecărei clase;

c) Să se afișeze elevii care au media generală mai mare decât media clasei.

<pre>select max(media), clasa from catalog group by clasa</pre>
<pre>select nume, media, (select avg(media) from CATALOG where c.clasa=clasa) as "Media clasei" from CATALOG c where media>(select avg(media) from CATALOG where c.clasa=clasa)</pre>
Se poate si fara partea verde, daca nu doriti afisarea mediei generale a clasei elevului respectiv

21. Pentru evidența produselor dintr-un magazin alimentar să se creeze o bază de date ce conține tabelul **PRODUSE** cu următoarele câmpuri: *cod, denumire, preț, cantitate, data_exp*.

- Să se adauge minim 5 înregistrări.
- Să se afișeze alfabetic produsele cu prețul mai mare decât un x citit de la tastatură
- Să se afișeze produsele expirate.

<pre>create table produse(cod number(3), denumire varchar2(30), pret number(6, 2), cantitate number(4), data_exp date)</pre>
<pre>insert into produse values (1, 'Carne', 25.43, 150, '15-May-2021') insert into produse values (2, 'Lapte', 5.50, 16, '15-Mar-2023') insert into produse values (3, 'Paine', 5, 10, '15-Jun-2021') insert into produse values (4, 'Peste', 23.25, 52, '15-Sep-2021') insert into produse values (5, 'Paste', 4.55, 100, '15-Oct-2021')</pre>
<pre>select * from produse where pret>:x order by denumire</pre>
<pre>select * from produse where data_exp<sysdate</pre>

22. Pentru evidența produselor dintr-un magazin alimentar să se creeze o bază de date ce conține tabelul **PRODUSE** cu următoarele câmpuri: *cod, denumire, preț, cantitate, data_exp*.

- Să se adauge minim 5 înregistrări.
- Să se caute în tabel produsul cu codul 25 și dacă acesta este găsit, să se mărească prețul acestuia cu 20.
- Să se afișeze produsele care au prețul mai mare decât prețul mediu.

<pre>create table produse(cod number(3), denumire varchar2(30), pret number(6, 2), cantitate number(4), data_exp date)</pre>
<pre>insert into produse values (1, 'Carne', 25.43, 150, '15-May-2021') insert into produse values (2, 'Lapte', 5.50, 16, '15-Mar-2023') insert into produse values (3, 'Paine', 5, 10, '15-Jun-2021') insert into produse values (4, 'Peste', 23.25, 52, '15-Sep-2021') insert into produse values (5, 'Paste', 4.55, 100, '15-Oct-2021')</pre>
<pre>update produse set preț = preț + 20 where cod=25</pre>
<pre>select denumire from produse where preț>(select AVG(preț)</pre>

from produse)

23. Pentru evidența cărților dintr-o bibliotecă să se creeze o bază de date ce conține tabelul **CĂRȚI** cu următoarele câmpuri: *cod, titlu, autor, editura, data_aparitie*

- Să se adauge minim 5 înregistrări.
- Să se afișeze cărțile scrise de autorul cu numele citit de la tastatură precum și numărul acestor cărți
- Să se afișeze alfabetic autorii care au publicat cărți în anul curent la o editură citită de la tastatură.

```
create table carti(cod number(3), titlu varchar2(100), autor varchar2(100), editura varchar2(50), data_aparitie date)
```

```
insert into carti values (1, 'Ion', 'Liviu Rebreanu', 'Art', '15-May-1920')
insert into carti values (2, 'Padurea spanzuratilor', 'Liviu Rebreanu', 'Art', '15-Mar-2021')
insert into carti values (3, 'Enigma Otiliei', 'George Calinescu', 'Paralela 45', '15-Jun-1938')
insert into carti values (4, 'Ultima noapte', 'Camil Petrescu', 'Polirom', '15-Sep-1930')
insert into carti values (5, 'Ciuleandra', 'Liviu Rebreanu', 'Polirom', '15-Oct-2021')
```

```
select titlu, (select count(*)
               from carti
               where autor=:a) "Numar de carti scrise"
from carti
where autor=:a
```

```
select autor
from carti
where extract(year from data_aparitie)=2022 and editura=:ed
order by autor
```

24. Pentru evidența cărților dintr-o bibliotecă să se creeze o bază de date ce conține tabelul **CĂRȚI** cu următoarele câmpuri: *cod, titlu, autor, editura, data_aparitie*

- Să se adauge minim 5 înregistrări.
- Să se afișeze, pentru fiecare autor, numărul titlurilor existente în bibliotecă
- Să se afișeze titlurile apărute la editura "RAO" între anii a1 și a2 citați de la tastatură.

```
create table carti(cod number(3), titlu varchar2(100), autor varchar2(100), editura varchar2(50), data_aparitie date)
```

```
insert into carti values (1, 'Ion', 'Liviu Rebreanu', 'RAO', '15-May-1920')
insert into carti values (2, 'Padurea spanzuratilor', 'Liviu Rebreanu', 'RAO', '15-Mar-2021')
insert into carti values (3, 'Enigma Otiliei', 'George Calinescu', 'RAO', '15-Jun-1938')
insert into carti values (4, 'Ultima noapte', 'Camil Petrescu', 'Polirom', '15-Sep-1930')
insert into carti values (5, 'Ciuleandra', 'Liviu Rebreanu', 'Polirom', '15-Oct-2021')
```

```
select autor, count(titlu)
from carti
group by autor
```

```
select titlu
from carti
where editura='RAO' and
extract(year from data_aparitie)<=:a2 and extract(year from data_aparitie)>=:a1
```

25. Pentru evidența jucăriilor dintr-un magazin pentru copii să se creeze o bază de date ce conține tabelul **JUCĂRII** cu următoarele câmpuri: *denumire, preț, categorie, numar_bucăți*.

- Să se adauge minim 5 înregistrări.
- Să se afișeze, jucăria cu cel mai mare preț din categoria "figurine"

c) Să se afișeze alfabetic denumirea jucăriilor pentru care numărul de bucăți este mai mic decât 5.

create table jucarii(denumire varchar2(100), pret number (6, 2), categorie varchar2(100), nrBuc number(3))
insert into jucarii values ('masina', 10, 'jocuri', 15) insert into jucarii values ('camion', 30, 'jocuri', 25) insert into jucarii values ('bicicleta', 100, 'jocuri', 4) insert into jucarii values ('toto', 10, 'figurine', 15) insert into jucarii values ('moto', 20, 'figurine', 15)
select denumire from jucarii where pret=(select max (pret) from jucarii where categorie='figurine') and categorie='figurine'
select denumire from jucarii where nrBuc<5 order by denumire

26. Pentru evidența jucăriilor dintr-un magazin pentru copii să se creeze o bază de date ce conține tabelul **JUCĂRII** cu următoarele câmpuri: *denumire, preț, categorie, număr_bucăți*.

- Să se adauge minim 5 înregistrări.
- Să se calculeze numărul de jucării care au prețul mai mic decât prețul mediu
- Să se afișeze toate jucăriile din aceeași categorie cu jucăria cu denumirea x citită de la tastatură

select count(*) from jucarii where pret < (select avg(pret) from jucarii)
select denumire from jucarii where categorie= (select categorie from jucarii where denumire=:x) and denumire!=:x

27. Pentru evidența cosmeticelor dintr-un magazin de profil, să se creeze o bază de date ce conține tabelul **COSMETICE** cu următoarele câmpuri: *cod, denumire, preț, categorie, data_exp*.

- Să se adauge minim 5 înregistrări.
- Să se șteargă din tabelă toate produsele expirate.
- Să se afișeze codurile cosmeticelor cu preț mai mare decât 30 precum și numărul total al acestor produse.

create table cosmetice(cod number(3), denumire varchar2(30), pret number(6, 2), categorie varchar2(40), data_exp date)
insert into cosmetice values (1, 'Crema1', 25.43, 'ingrijire corp', '15-May-2021') insert into cosmetice values (2, 'Crema2', 5.50, 'ingrijire corp', '15-Mar-2023') insert into cosmetice values (3, 'Crema3', 5, 'ingrijire corp', '15-Jun-2021') insert into cosmetice values (4, 'Crema4', 23.25, 'ingrijire fata', '15-Sep-2021') insert into cosmetice values (5, 'Crema5', 4.55, 'ingrijire fata', '15-Oct-2021')
delete from cosmetice where data_exp<sysdate

```

select cod, (select count(*)
              from cosmetice
              where pret>30) as "Numar total"
from cosmetice
where pret>30

```

28. Pentru evidența cosmeticelor dintr-un magazin de profil, să se creeze o bază de date ce conține tabelul **COSMETICE** cu următoarele câmpuri: *cod*, *denumire*, *preț*, *categorie*, *data_exp*.

a) Să se adauge minim 5 înregistrări.

b) Să se afișeze denumirea și codul cosmeticelor din categoria X, descrescător după preț

c) Să se afișeze categoria/categoriile în care există produse cu preț mai mic decât prețul minim din categoria "machiaj".

```

insert into cosmetice values (5, 'Crema6', 15.25, 'machiaj', '15-May-2021')

```

```

select denumire, cod
from cosmetice
where categorie=:x
order by pret desc

```

```

select categorie
from cosmetice
where pret < (select min(pret)
              from cosmetice
              where categorie='machiaj')

```

29. Pentru evidența punctajelor concurenților la probele dintr-un concurs sportiv, să se creeze o bază de date ce conține tabelul **CONCURENȚI** cu următoarele câmpuri: *cod*, *nume*, *club_sportiv*, *punctaj_p1*, *punctaj_p2*, *punctaj_p3*

a) Să se adauge minim 5 înregistrări.

b) Să se afișeze concurenții cu media punctajelor mai mare decât media concurentului cu codul x citit de la tastatură

c) Să se afișeze numele și clubul sportiv al concurenților care au obținut cel mai mare punctaj la proba cu numărul 3 (punctaj p3).

```

create table concurenti(cod number(3), nume varchar2(30), club varchar2(40),p1 number(3),p2 number(3),
p3 number(3) )

```

```

insert into concurenti values (1, 'Popescu', 'C1', 10, 10, 6)
insert into concurenti values (2, 'Ionescu', 'C2', 8, 6, 6)
insert into concurenti values (3, 'Georgescu', 'C3', 10, 9, 8)
insert into concurenti values (4, 'Stanescu', 'C1', 10, 5, 6)
insert into concurenti values (5, 'Vasilescu', 'C2', 7, 10, 6)

```

```

select nume
from concurenti
where (p1+p2+p3)/3>(select (p1+p2+p3)/3
                    from concurenti
                    where cod=:x)

```

```

select nume, club
from concurenti
where p3 = (select max(p3)
            from concurenti)

```

30. Pentru evidența punctajelor concurenților la probele dintr-un concurs sportiv, să se creeze o bază de date ce conține tabelul **CONCURENȚI** cu următoarele câmpuri: *cod*, *nume*, *club_sportiv*, *punctaj_p1*, *punctaj_p2*, *punctaj_p3*

a) Să se adauge minim 5 înregistrări.

b) Să se afișeze concurenții clubului sportiv cu numele citit de la tastatură, descrescător după *punctaj_p1*

c) Să se afișeze concurenții care au cel puțin un punctaj nul.

insert into concurenti values (6, 'protopopescu', 'C2', 7, 9, 6)
--

select nume from concurenti where club=:x order by p1 desc

select nume from concurenti where p1 is null or p2 is null or p3 is null
--