

Subiectul 1.

Se consideră o bază de date cu 2 tabele: FURNIZORI (id_furnizor, denumire, oras); PRODUSE (id_produș, denumire, pret, data_expirarii, id_furnizor)

a) Creați tabelele și populați-le cu date pertinente;

```
CREATE TABLE furnizori (id_furnizor NUMBER(5) PRIMARY KEY,denumire VARCHAR2(20),oras VARCHAR2(20))
```

```
CREATE TABLE produse (id_produș NUMBER(5) PRIMARY KEY,denumire VARCHAR2(20),pret NUMBER(7,2),data_expirarii DATE,id_furnizor NUMBER(5),FOREIGN KEY (id_furnizor) REFERENCES furnizori(id_furnizor))
```

```
INSERT INTO furnizori VALUES(1,'S.C. ALFA SRL','Sibiu')
```

```
INSERT INTO furnizori VALUES(2,'S.C. BETA SRL','Rm. Valcea')
```

```
INSERT INTO furnizori VALUES(3,'S.C. GAMA SRL','Sibiu')
```

```
INSERT INTO produse VALUES(1,'Pasta de dinti',12,'jan-12-2019',2)
```

```
INSERT INTO produse VALUES(2,'Periuta de dinti',35.50,'jan-25-2019',2)
```

b) Afișați produsul/produsele cu preț minim;

```
SELECT denumire FROM produse WHERE pret=(SELECT MIN(pret) FROM produse)
```

c) Afișați produsele ce expiră în luna curentă;

```
SELECT * FROM produse WHERE EXTRACT(YEAR FROM data_expirarii) = EXTRACT(YEAR FROM SYSDATE) AND EXTRACT(MONTH FROM data_expirarii) = EXTRACT(MONTH FROM SYSDATE)
```

d) Afișați produsele ce provin de la furnizori din Rm.Vâlcea.

```
SELECT p.denumire, p.pret, p.data_expirarii, f.denumire FROM produse p JOIN furnizori f ON(p.id_furnizor=f.id_furnizor) WHERE f.oras = 'Rm. Valcea' OR f.oras = 'Valcea'
```

Subiectul 2.

Se consideră o bază de date cu 2 tabele: EDITURI (id_editură, denumire, adresă) CARTI (id_carte, titlu, autor, id_editură, anul_apariției). Dacă o carte are mai mulți autori se va reține coordonatorul.

a) Creați tabelele și populați-le cu date pertinente;

```
CREATE TABLE edituri (id_editura NUMBER(5) PRIMARY KEY,denumire VARCHAR2(30),adresa VARCHAR2(25))
```

```
CREATE TABLE carti (id_carte NUMBER(10),autor VARCHAR2(25),id_editura NUMBER(5),FOREIGN KEY (id_editura) REFERENCES edituri(id_editura),anul_aparitei NUMBER(5))
```

```
INSERT INTO edituri VALUES (1,'Didactica','Bucuresti, Sectorul 1')
```

```
INSERT INTO edituri VALUES (2,'ALFA','Sibiu, Cibirului 1')
```

```
INSERT INTO edituri VALUES (3,'BETA','Rm. Valcea, Dacia 12')
```

```
INSERT INTO edituri VALUES (4,'HUMANITAS', 'Bucuresti, Luceafarul 14')
```

```
INSERT INTO edituri VALUES (5,'GAMA', 'SIBIU')
```

```
INSERT INTO edituri VALUES (6,'OMEGA', 'SIBIU')
```

```
INSERT INTO carti VALUES (1,'Mihai Eminescu',4,2019)
```

```
INSERT INTO carti VALUES (2,'Ion Creanga',4,2018)
```

```
INSERT INTO carti VALUES (3,'Mihai Bold',1,2019)
```

```
INSERT INTO carti VALUES (4,'Bogdan Popescu',2,2019)
```

b) Afișați cartea/cărțile apărute în anul curent;

```
SELECT * FROM carti WHERE anul_aparitei=TO_CHAR(SYSDATE, 'YYYY')
```

c) Afișați editura/editurile ce au adresa în Sibiu;

```
SELECT * FROM edituri WHERE adresa='Sibiu'
```

d) Afișați editura/editurile ce au un număr maxim de cărți.

```
SELECT * FROM edituri WHERE id_editura IN (SELECT id_editura FROM carti GROUP BY id_editura HAVING COUNT(*) = (SELECT MAX(COUNT(*)) FROM carti GROUP BY id_editura))
```

Subiectul 3.

Să se realizeze o interfață pentru gestionarea cărților dintr-o bibliotecă. Se utilizează o bază de date cu două tabele având următoarele structuri: tabela CARTI cu structura: ISBN, COD_AUTOR, TITLUL, ANUL_A, NR_VOL, PRET și tabela AUTORI: COD_AUTOR, NUME, PRENUME, DATA_N, LOCALITATEA_N. Să se creeze baza de date și tabelele, introducând 10 înregistrări.

Cerințe:

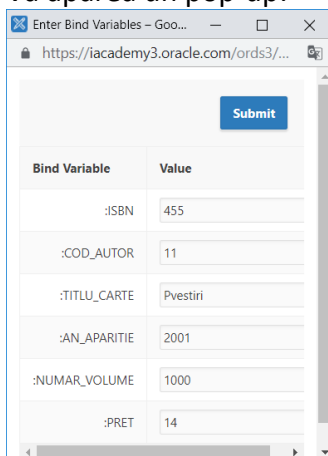
a) Introduceți o nouă carte în evidența bibliotecii, citind datele de la tastatură;

```
CREATE TABLE autori (cod_autor NUMBER(5) PRIMARY KEY, nume VARCHAR2(15), prenume VARCHAR2(25), data_n DATE, localitatea_n VARCHAR2(20))
```

```
CREATE TABLE carti3 (isbn VARCHAR2(20) PRIMARY KEY, cod_autor NUMBER(5), FOREIGN KEY (cod_autor) REFERENCES autori(cod_autor), titlul VARCHAR(25), anul_a VARCHAR(4), nr_vol NUMBER(5), pret NUMBER(7,2))
```

```
INSERT INTO carti3 VALUES (:isbn, :Cod_autor, :Titlu_carte, :An_aparitie, :Numar_volum, :Pret)
```

Va apărea un pop-up:



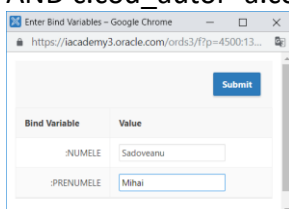
Bind Variable	Value
:ISBN	455
:COD_AUTOR	11
:TITLU_CARTE	Pvestiri
:AN_APARTIE	2001
:NUMAR_VOLUM	1000
:PRET	14

b) Afișați cărțile care au numărul maxim de volume;

```
SELECT * FROM carti3 WHERE nr_vol IN (SELECT nr_vol FROM carti3 GROUP BY nr_vol HAVING nr_vol=(SELECT MAX(nr_vol) FROM carti3))
```

c) Afișați cărțile existente în bibliotecă care au autorul cu numele și prenumele citit de la tastatură.

```
SELECT c.titlul FROM carti3 c, autori a WHERE a.num=:numele AND a.prenume=:prenumele AND c.cod_autor=a.cod_autor
```



Bind Variable	Value
:NUMELE	Sadoveanu
:PRENUMELE	Mihai

Subiectul 4.

Să se realizeze o interfață utilă pentru o farmacie. Se utilizează o bază de date cu două tabele având următoarele structuri: tabela **Medicamente**: ID, DENUMIRE, COMPENSAT, ID_PROD, PRET și tabela **Producători**: cu structura ID_PROD, NUME_PROD, ADRESA, ȚARA. Să se creeze baza de date și tabelele, introducând 10 înregistrări.

```
CREATE TABLE producatori4 (id_prod NUMBER(5) PRIMARY KEY,nume_prod VARCHAR(25),adresa VARCHAR2(25),tara VARCHAR2(20))
```

```
CREATE TABLE medicamente4 (id NUMBER(5) PRIMARY KEY,denumire VARCHAR2(15),compensat NUMBER(1),id_prod NUMBER(5),FOREIGN KEY (id_prod) REFERENCES producatori(id_prod),pret NUMBER(7,2))
```

```
INSERT INTO producatori4 VALUES (1,'S.C. ALFA SRL','Str. Dacia 20 Rm. Valcea','Romania')
```

```
INSERT INTO producatori4 VALUES (2,'S.C. BETA SRL','Str. Livezi 2 Rm. Valcea','Romania')
```

```
INSERT INTO medicamente4 VALUES (1,'Piramidon',0,1,5.75)
```

```
INSERT INTO medicamente4 VALUES (2,'Carmol',1,2,4.50)
```

```
INSERT INTO medicamente4 VALUES (3,'Spirt',0,1,8.50)
```

Cerințe:

a) Modificați prețul unui medicament, cu denumirea citită de la tastatură;

```
UPDATE medicamente4 SET pret=:pret WHERE denumire=:denumire
```

b) Listați medicamentele care nu beneficiază de compensare;

```
SELECT * FROM medicamente4 WHERE compensat=0
```

c) Afișați toate medicamentele din farmacie produse de către un producător citit de la tastatură.

```
SELECT m.id,m.denumire,m.compensat,m.id_prod,m.pret FROM medicamente4 m JOIN producatori4 p ON (m.id_prod=p.id_prod) WHERE p.nume_prod=:nume
```

Subiectul 5.

Să se realizeze o interfață pentru gestionarea unei colecții de albume muzicale. Se utilizează o bază de date cu următoarele tabele: tabela Discuri ID, ID_ARTIST, ALBUM, AN_AP, SUPORT și tabela Artiști cu structura: ID_ARTIST, NUME, PRENUME. Să se creeze baza de date introducând 10 înregistrări.

```
CREATE TABLE artisti5 (id_artist NUMBER(5) PRIMARY KEY,nume VARCHAR(25),prenume VARCHAR(25))
```

```
CREATE TABLE discuri5 (id NUMBER(5) PRIMARY KEY,id_artist NUMBER(5),FOREIGN KEY (id_artist) REFERENCES artisti5(id_artist),album VARCHAR2(30),an_ap VARCHAR2(4),suport VARCHAR2(20))
```

```
INSERT INTO artisti5 VALUES (1,'Anghel','Monica')
```

```
INSERT INTO artisti5 VALUES (2,'Bitman','Dan')
```

```
INSERT INTO artisti5 VALUES (3,'Cotabita','Gabriel')
```

```
INSERT INTO discuri5 VALUES (1,1,'ALFA','2000','DVD')
```

```
INSERT INTO discuri5 VALUES (2,1,'BETA','2005','DVD')
```

Cerințe:

a) Afișați toate albumele care sunt imprimate pe DVD;

```
SELECT * FROM discuri5 WHERE suport='DVD'
```

b) Adăugați un album nou al artistului x, cu nume și prenume citit de la tastatură;

```
INSERT INTO discuri5 VALUES ((SELECT MAX(id)+1 FROM discuri5),(SELECT id_artist FROM artisti5 WHERE nume=:nume),'Nebunia lui Fuego',2016,'DVD')
```

c) Afișați în ordinea apariției (AN_AP), toate albumele unui artist specificat.

```
SELECT DISTINCT d.album,d.an_ap,d.suport FROM discuri5 d JOIN artisti5 a ON d.id_artist = (SELECT id_artist FROM artisti5 WHERE nume=:nume AND prenume=:prenume) ORDER BY an_ap
```

Subiectul 6.

Să se realizeze o interfață pentru gestionarea mărfurilor dintr-un depozit de calculatoare. Se utilizează o bază de date cu următoarele tabele: Produse: COD_PROD, DENUMIRE, UM, CANTIT și tabela Producători: COD_PROD, NUME, ADRESA, TELEFON.

```
CREATE TABLE produse6 (cod_prod NUMBER(5) PRIMARY KEY,denumire VARCHAR(25),um  
VARCHAR2(10),cantit NUMBER(7,2))
```

```
CREATE TABLE producatori6 (cod_prod NUMBER(5) PRIMARY KEY,nume VARCHAR2(10),adresa  
VARCHAR(25),telefon VARCHAR(15))
```

Cerințe:

a) Afișați toate mărfurile cu stoc 0;

```
SELECT denumir FROM Produse6 WHERE cantit=0
```

b) Introduceți noi produse ale unui nou producător, cu numele citit de la tastatură;

```
INSERT INTO produse6 (cod_prod,denumir,cantit,um) VALUES ('6', 'Carti', '23', 'buc')
```

c) Afișați numele producătorilor care produc tastaturi.

```
SELECT nume FROM produse6 WHERE denumir='Tastaturi'
```

Subiectul 7.

În tabăra cercetașilor din acest an au participat tineri din 6 județe. Fiecare județ a fost reprezentat de cel puțin o unitate de exploratori și cel puțin o unitate de temerari. Fiecare unitate este formată din doi cercetași. În cadrul concursului taberei au fost organizate 3 probe: tir, trasul cu arcul și cățărare. Pentru fiecare participant organizatorii trebuie să rețină următoarele date: numele și prenumele, județul, numele unității de cercetași din care provin, categoria (explorator sau temerar), punctajul obținut (cuprins între 0 și 10 puncte) la fiecare dintre cele trei probe din cadrul concursului. Organizatorii pregătesc festivitatea de premiere și trebuie să obțină răspunsuri cât mai rapid la următoarele întrebări:

a) Ce județ a câștigat concursul taberei? (se acumulează punctajele unităților din fiecare județ)

```
SELECT judet FROM unitati WHERE (p_arc+p_catarare+p_tir)=(SELECT  
MAX(p_arc+p_catarare+p_tir) FROM unitati)
```

b) Care sunt numele cercetașilor care au avut punctajele cele mai mari pentru fiecare probă în parte?

```
SELECT a.nume FROM participant a, unitati b WHERE a.nume_unitate=b.nume_unitate AND  
p_tir=(SELECT MAX(p_tir) FROM unitati) AND p_arc=(SELECT max(p_arc) FROM unitati) AND  
p_catarare=(SELECT max(p_catarare) FROM unitati)
```

c) Care este județul cu cele mai multe unități participante în tabără?

```
SELECT judet FROM unitati GROUP BY judet HAVING COUNT(nume_unitate) = (SELECT  
MAX(COUNT(nume_unitate)) FROM unitati GROUP BY judet)
```

Propuneți un model de baze de date (tabele, relații etc.) și adăugați înregistrări în tabele astfel încât organizatorii concursului să găsească mai ușor răspunsul la întrebările formulate.

Subiectul 8.

La un concurs de dans perechile de dansatori se pot înregistra la una din cele două probe: salsa sau tango. Știind că participanții pot fi din mai multe orașe, realizați un model de baze de date astfel încât juriul să poată ține evidența participanților împreună cu punctajele obținute la probele de dans (între 0 și 10).

Răspundeți la următoarele cerințe:

a) Afișați clasamentul perechilor pentru fiecare probă;

```
SELECT echipa,nume,prenume FROM dansatori ORDER BY proba,scor DESC
```

b) Care este orașul din care au fost înscriși cei mai mulți dansatori?

```
SELECT oras FROM dansatori GROUP BY oras HAVING COUNT(oras) = (SELECT MAX(count(*))  
FROM dansatori GROUP BY oras )
```

c) Care dintre cele două probe de dans a avut participanți mai mulți?

```
SELECT proba FROM dansatori GROUP BY proba HAVING COUNT(proba) = (SELECT  
MAX(count(*)) FROM dansatori GROUP BY proba)
```


Subiectul 9.

Ioana Popescu a preluat librăria „Cartea fantastică” de la mătușa sa care s-a pensionat anul trecut. Deoarece vânzările au crescut destul de mult, Ioana nu mai poate continua afacerea ca până acum și are nevoie de o bază de date în care să țină inventarul cărților și situația vânzărilor: autor, titlul cărții, anul apariției, editura, numărul exemplarelor, prețul cărții, categorie (SF, poezie, biografie, tehnică etc.), data la care s-a efectuat vânzarea, cărțile vândute, cantitatea. Ajutați-o pe Ioana să construiască o bază de date care să-i permită administrarea librăriei. Construiți 3 întrebări la care Ioana trebuie să răspundă (de exemplu: căutarea după autori, titluri de cărți, editura sau tipuri de literatură, clasamentul editurilor cu cele mai multe cărți din librărie, autorul cu cele mai mari vânzări etc.)

a)

```
SELECT * FROM inventar WHERE categorie ='Poezie'
```

b)

```
SELECT * FROM Inventar WHERE pret>20
```

c)

```
SELECT autor FROM inventar WHERE anul_aparitiei<2000
```

Subiectul 10.

Să se realizeze o bază de date capabilă să gestioneze produsele dintr-un magazin de piese auto care cumpără de la diverși producători interni piese și echipamente. Baza de date va răspunde următoarelor cerințe:

a) Afișați producătorii din orașul x, citit de la tastatură, de la care firma cumpără produse;

```
SELECT nume_producator FROM producatori WHERE oras=:o AND vanzari>0
```

b) Afișați toate produsele care au preț maxim și se află în stoc de mai mult de un an;

```
SELECT nume FROM magazine WHERE (SYSDATE-365>data_orimire) AND pret=(SELECT  
MAX(pret) FROM magazine)
```

c) Micșorați prețul produselor achiziționate de la producătorii din București cu 5% și afișați pe ecran noile prețuri.

```
SELECT pret*0.95 AS "Pret Nou" FROM magazine,producatori
```

Subiectul 11.

Să se creeze tabelele necesare memorării informațiilor despre angajații unei societăți, organizați ierarhic pe departamente. Fiecare departament trebuie să aibă cel puțin un angajat. Fiecare departament are un șef.

```
CREATE TABLE departamente11 (id NUMBER(5) PRIMARY KEY,nume_d VARCHAR2(20))
INSERT INTO departamente11 VALUES (1,'Productie')
INSERT INTO departamente11 VALUES (2,'Desfacere')
INSERT INTO departamente11 VALUES (3,'Resurse umane')
INSERT INTO departamente11 VALUES (4,'Motoare electrice')
CREATE TABLE angajati11 (id_a NUMBER(5) PRIMARY KEY,nume VARCHAR2(20),prenume
VARCHAR2(25),id_d NUMBER(5), FOREIGN KEY (id_d) REFERENCES departamente11(id))
INSERT INTO angajati11 VALUES (1,'Pop','Viorel',1)
INSERT INTO angajati11 VALUES (2,'Popescu','Dan',1)
INSERT INTO angajati11 VALUES (3,'Motrescu','Vlad',1)
INSERT INTO angajati11 VALUES (4,'Sandu','Daniel',1)
INSERT INTO angajati11 VALUES (5,'Popa','Andrei',2)
INSERT INTO angajati11 VALUES (6,'Popian','Dorin',2)
INSERT INTO angajati11 VALUES (7,'Sararu','Mihai',2)
INSERT INTO angajati11 VALUES (8,'Angelescu','Ion',2)
INSERT INTO angajati11 VALUES (9,'Georgescu','Nicolae',2)
```

Se cere:

a) Să se afișeze departamentele al căror nume începe cu 'M' sau 'P';

```
SELECT nume_d FROM departamente11 WHERE nume_d LIKE 'M%' OR nume_d LIKE 'P%'
```

b) Să se afișeze toți angajații grupați pe departamente, ordonați alfabetic după nume;

```
SELECT d.nume_d,a.nume,a.prenume FROM angajati11 a,departamente11 d ORDER BY
a.id_d,a.nume,a.prenume
```

c) Să se afișeze departamentele care au doar x angajați, x citit de la tastatură.

```
SELECT d.nume_d FROM angajati11 a,departamente11 d WHERE a.id_d=d.id GROUP BY
d.nume_d HAVING (COUNT(d.id)=:Nr_departament)
```

Subiectul 12

Creați baza de date călătorii formată din tabelele destinații (IDD, DENUMIRE, ȚARA, DESCRIERE) și clienți (COD_C, NUME, PRENUME, ADRESA, TELEFON) și un al treilea tabel care să rețină preferințele clienților în materie de destinații. Construiți tabelele și completați cu minim 3 înregistrări tabela destinații și minim 5 tabela clienți.

a) Afișați clienții cu numele x, citit de la tastatură;

```
SELECT *FROM clienti WHERE nume=:enter_nume
```

b) Afișați destinațiile din România;

```
SELECT * FROM destinatii WHERE tara='Romania'
```

c) Afișați pentru fiecare destinație, numărul clienților, care au preferat-o.

```
SELECT COUNT(cod_c) FROM clienti JOIN destinatii d ON (client.cod_c=d.idd) GROUP BY denumire
```

Subiectul 13.

Creați o bază de date care să păstreze informațiile despre locatarii unei asociații de proprietari și care să răspundă următoarelor interogări:

a) Numărul apartamentelor care au restanțe în luna curentă;

```
SELECT apartament FROM locatari WHERE restante>0
```

b) Care este familia cu restanțe ce depășesc o sumă x dată și care este numele membrilor săi;

```
SELECT a.nume, a.membru1, a.membru2, a.membru3 FROM familie a, locatari b WHERE  
a.nume=b.nume AND restante>11
```

c) Să se afișeze locatarii și numărul apartamentului lor, care lucrează în același loc de muncă cu cel citit de la tastatură.

```
SELECT nume, apartament FROM locatari WHERE (loc_de_munca=(SELECT loc_de_munca FROM  
locatari WHERE nume=:nume))
```

Subiectul 14. (ei zic ca nu merge verificati)

Se dorește realizarea unei baze de date cu evidența operațiilor realizate de o casă de schimb valutar, ce folosește următoarele tipuri de monede: euro, dolar american, dolar australian, dolar canadian, liră sterlină, yen japonez etc. Din acest motiv, se oferă următoarele informații:

- în cadrul casei de schimb valutar se fac operații de „vânzare” și „cumpărare” valută, cu un comision de vânzare și un comision de cumpărare la un curs stabilit de Banca Națională a României la data respectivă;

- pentru o operație realizată la casă, sunt necesare următoarele date: codul personal al clientului, numele și prenumele acestuia, tipul operației (vânzare sau cumpărare de valută), data schimbului valutar, suma în valută, tipul monedei (euro, dolar american, dolar australian, dolar canadian, liră sterlină, yen japonez etc.), valoarea în lei a monedei (conform cursului valutar al BNR din ziua respectivă).

Pentru o bună comunicare cu angajații și clienții, să se realizeze o bază de date cu cel puțin 2 tabele relaționate, care să permită obținerea următoarelor informații referitoare la activitatea casei de schimb valutar:

a) Care sunt operațiile de schimb valutar ce s-au făcut pentru fiecare tip de monedă (euro, dolar american, dolar australian, dolar canadian, liră sterlină, yen japonez. etc.)?

```
SELECT o.id_client,o.numele_clientului,o.prenumele_clientului,o.tipul_operatiei,o.suma FROM operatii o JOIN monede m on(o.id_moneda=m.id_moneda) GROUP BY m.moneda
```

b) Ce operații de vânzare de dolari americani, în valoare mai mare de 1000 de lei s-au făcut acum 10 zile?

```
SELECT o.id_client,o.nume,o.prenume,o.operatie,o.suma from operatii o JOIN moneda m ON (o.id_moneda=m.id_moneda) WHERE m.tipul_operatiei='Vanzare' AND (o.suma>1000 AND EXTRACT(DAY from SYSDATE)-EXTRACT(DAY FROM SYSDATE)=0)
```

c) Actualizați baza de date știind că pentru operațiile de „vânzare” realizate în luna curentă se dorește majorarea comisionului cu 0,01%.

```
UPDATE operatii SET suma=suma*0.01/100 WHERE tipul_operatiei='Vanzare'
```

d) Numărul total de operații valutare și valoarea totală a comisionului perceput pentru acestea.

Problema 15.

Proprietarul unui depozit de produse alimentare își propune să țină evidența fructelor și legumelor pe care le are pe stoc la un moment dat. Din acest motiv, pentru realizarea bazei de date se oferă următoarele informații:

- furnizorii depozitului sunt firme consacrate pe piață, care au dovedit seriozitate și promptitudine în livrarea comenzilor, de aceea depozitul le oferă exclusivitate: Hermes SFA, Asil2000, SC Laura22 SRL, Bel Company, Alex Star etc.

- în depozit se află legume și fructe proaspete, congelate sau sub formă de conservă, ambalate în cutii, borcane și sticle, navete sau saci;

- pentru fiecare produs se oferă următoarele date: tipul produsului (fructe sau legume), forma (proaspăt, congelat, uscat sau sub formă de conservă), tipul ambalajului, producătorul, distribuitorul, data fabricației, data expirării, cantitatea, prețul pe unitatea de măsură și alte informații (condiții de depozitare, de transport etc.);

Pentru o bună comunicare cu angajații și clienții depozitului, proprietarul dorește realizarea unei baze de date cu cel puțin 2 tabele relaționate, din care să poată obține următoarele informații:

```
CREATE TABLE producatori15 (idprod NUMBER(5) PRIMARY KEY, denprod VARCHAR2(25))
```

```
INSERT INTO producatori15 VALUES (1, 'SC ALFA SA')
```

```
INSERT INTO producatori15 VALUES (2, 'SC BETA SA')
```

```
INSERT INTO producatori15 VALUES (3, 'SC GAMA SA')
```

```
CREATE TABLE produse15 (idp NUMBER(5) PRIMARY KEY, denp VARCHAR2(25), cantp NUMBER(7,2), pretp NUMBER(7,2), ambalaj VARCHAR2(20), dataf DATE, dataexp DATE, idprod NUMBER(5), FOREIGN KEY (idprod) REFERENCES producatori15(idprod))
```

```
INSERT INTO produse15 VALUES (1, 'Mere', 125, 2.75, 'lada', '25/oct/2018', '15/may/2019',1)
```

```
INSERT INTO produse15 VALUES (2, 'Pere', 125, 3.99, 'lada', '20/oct/2018', '15/apr/2019',1)
```

```
INSERT INTO produse15 VALUES (3, 'Gem prune', 1500, 9.75, 'sticla', '30/oct/2018', '20/jul/2021',2)
```

```
INSERT INTO produse15 VALUES (4, 'Gem gutui', 1500, 9.75, 'sticla', '23/nov/2016', '20/feb/2019',2)
```

```
INSERT INTO produse15 VALUES (5, 'Gem afine', 1500, 12.75, 'sticla', '23/aug/2015', '20/feb/2018',2)
```

```
INSERT INTO produse15 VALUES (6, 'Gem caise', 35, 10.99, 'sticla', '23/aug/2015', '20/feb/2018',2)
```

```
INSERT INTO produse15 VALUES (7, 'Caise', 20, 4.99, 'sticla', '23/aug/2015', '20/feb/2019',3)
```

```
INSERT INTO produse15 VALUES (8, 'Gem caise', 20, 4.99, 'lada', '23/dec/2018', '20/feb/2023',3)
```

a) De ce legume și fructe dispune depozitul pentru fiecare furnizor (Hermes SFA, Asil2000, SC Laura22 SRL, Bel Company, Alex Star etc.)?

```
SELECT a.denp, b.denprod FROM produse15 a, producatori15 b WHERE (a.idprod=b.idprod)
```

```
AND (denprod=:Producator)
```

b) Ce conserve în ambalaj de sticlă în cantități mai mari decât 50 se găsesc în depozit?

```
SELECT * FROM produse15 WHERE ambalaj='sticla' AND cantp>=50
```

c) Actualizați baza de date știind că pentru legumele și fructele proaspete, cu data fabricației în lunile de iarnă (decembrie, ianuarie și februarie), prețurile se măresc cu 15%.

```
UPDATE produse15 SET pretp=1.15*pretp WHERE dataf BETWEEN '01/dec/2018' AND '28/feb/2019'
```

d) Numărul total de produse din stoc care sunt disponibile clienților și nu au expirat încă.

```
SELECT COUNT(*) AS "Numar total prpduse" FROM produse15 WHERE cantp!=0 AND dataexp>SYSDATE
```

Subiectul 16.

Conducerea unei companii aeriene internaționale dorește realizarea unei baze de date cu rezervările pentru zborurile ce au loc cu avioanele deținute de această companie, cu plecare din aeroporturile internaționale: Henri Coandă din București, Leonardo da Vinci din Roma, Gatwick din Londra, Charles de Gaulle din Paris etc. Din acest motiv, pentru realizarea bazei de date sunt oferite următoarele informații:

- în regulamentul companiei aeriene există următoarele prevederi: se permite unui avion să facă cel mult 2 curse săptămânal, în două zile diferite (luni, marți, miercuri, joi, vineri, sâmbătă sau duminică), cu pauză de cel puțin 2 zile între ele; nu se fac mai mult de 2 zboruri din același aeroport către aceeași destinație într-o săptămână;
- pentru fiecare avion se dau următoarele date: codul avionului, tipul avionului (Airbus A380, Boeing B737, MD-82, Wing 767 etc.), capacitatea (numărul maxim de pasageri) pe fiecare clasă;
- pentru fiecare zbor planificat se cunosc următoarele: numărul cursei, aeroportul din care se face plecarea, destinația (Berlin, București, Budapesta, Chișinău, Londra, Moscova, Paris, Roma, Sofia) și aeroportul în care se aterizează,, data și ora plecării, data și ora sosirii, ziua din săptămână, codul avionului;
- la rezervarea unui loc la un anumit zbor se solicită următoarele date: codul personal al pasagerului, numele și prenumele, numărul locului în avion, clasa (1, 2 sau „business”), numărul cursei. Pentru o bună informare a angajaților și a pasagerilor companiei aeriene, conducerea acesteia dorește realizarea unei baze de date cu cel puțin 2 tabele relaționate, care să permită obținerea următoarelor informații:

a) Care sunt rezervările ce s-au făcut pentru fiecare aeroport cu care colaborează compania aeriană (Henri Coandă, Leonardo da Vinci, Gatwick, Charles de Gaulle etc.)?

b) Ce pasageri zboară cu avioane de tipul Airbus A380, cu capacitatea la clasa „business” mai mică decât 50, cu plecarea din aeroportul Charles de Gaulle, în ziua de marți?

c) Care este prețul unui bilet de la București la Londra, la o cursă din luna decembrie, la clasa 1, știind că se oferă o reducere de preț de 15% pentru toate zborurile planificate în perioada 1-15 decembrie?

d) Numărul de rezervări și valoarea totală încasată pentru zborurile cu destinația București.

Subiectul 17.

Proprietarul unui complex turistic își propune să țină evidența hotelurilor, pensiunilor și cabanelor pe care le oferă amatorilor de excursii și sporturi montane. Din acest motiv, se oferă următoarele informații pentru realizarea bazei de date:

- în incinta complexului se află hoteluri de 2 și 3 stele, pensiuni de 2 și 3 margarete, o pârtie înconjurată de cabane calificate cu 1 sau 2 stele;
- fiecare unitate de cazare are între 5 și 50 de locuri de cazare grupate în camere de tipul: single (1 loc), double (2 locuri), multiple (3, 4 sau mai multe locuri) și apartamente;
- pentru fiecare cameră se oferă următoarele date: dotările (telefon, televizor, frigider, balcon, aer condiționat), număr locuri de cazare, prețul pe cameră, dacă este liberă sau ocupată și alte informații (vederea, poziția față de pârtie etc.).

Pentru o bună comunicare cu angajații și clienții complexului turistic, proprietarul dorește realizarea unei baze de date cu cel puțin 2 tabele relaționate, din care să poată obține următoarele informații:

a) De ce camere dispune complexul pentru fiecare tip de unitate de cazare (hotel, pensiune sau cabană)?

```
SELECT a.pret-a.pret*0.1 FROM camere a,cazare b WHERE a.id_cam=b.id_cam AND b.complex='Hotel' AND b.calitate='2 stele'
```

b) Ce camere de 2 locuri cu balcon sunt libere în hotelurile de 3 stele?

```
SELECT SUM(nr_loc_caz) FROM camere WHERE ocupare='nu'
```

c) Să se actualizeze prețurile camerelor, știind că în lunile de vară (iunie, iulie și august) se face o reducere de 10% pentru camerele de 2 stele din hotelurile complexului.

```
SELECT a.id_cam FROM camere a,cazare b WHERE a.ocupare='nu' AND a.id_cam=b.id_cam AND a.nr_loc_caz=2 AND a.balcon='da'AND b.complex='Hotel'AND b.calitate='3 stele'
```

d) Numărul total de locuri de cazare disponibile noilor clienți.

```
SELECT a.id_cam,b.complex FROM camere a,cazare b WHERE a.ocupare='nu' AND a.id_cam=b.id_cam
```

Subiectul 18.

Organizatorii Balului Bobocilor din acest an s-au hotărât ca să gestioneze concursul mult mai ușor. Ei trebuie să construiască o bază de date în care să rețină numele și prenumele participanților, sexul (M/F), precum și punctajele (între 0 și 10) obținute la probele concursului (dans, talent, cultura generală, surpriză etc.). Construiți un model de baze de date care să conțină cel puțin 2 tabele relaționate din care să se poată extrage următoarele informații:

a) Care sunt câștigătorii din acest an ai concursului Miss& Mister Boboc?

```
SELECT nume,prenume FROM concurenti WHERE punctaj=(SELECT MAX(punctaj) FROM concurenti)
```

b) Care este clasamentul la proba de dans?

```
SELECT a.nume,a.prenume,b.punctaj FROM participanti a,probe b WHERE a.id_conc=b.id_conc AND b.probe_conc='dans' ORDER BY b.punctaj DESC
```

c) Numele concurentului/concurenților care au obținut cel mai mare punctaj precum și proba/probele la care a/au fost obținut acest punctaj maxim.

```
SELECT a.nume,a.prenume,b.punctaj FROM participanti a,probe b WHERE a.id_conc=b.id_conc AND b.punctaj=(SELECT MAX( b.punctaj) FROM probe b)
```

Subiectul 19.

La examenul de bacalaureat se trec în baza de date următoarele informații: Nume, Prenume, Nota_Rom, Nota_disciplina_profil, Nota_disciplina_alegere.

Cerințe:

a) Afișarea elevilor admiși în ordinea descrescătoare a mediilor;

```
SELECT * FROM examene WHERE nota_rom>=5 AND nota_disciplina_profil>=5 AND  
nota_disciplina_alegere>=5 AND medie>=6 ORDER BY medie DESC
```

b) Afișarea elevilor respinși în ordine alfabetică;

```
SELECT * FROM examene WHERE medie
```

c) Afișați media aritmetică a mediilor elevilor admiși. Pentru a fi admis, fiecare notă trebuie să fie mai mare sau egală cu 5 și media trebuie să fie cel puțin 6.

```
SELECT ROUND(AVG(medie),2) FROM examene WHERE medie>=6
```

Subiectul 20.

La o dezbatere despre educație participă elevi și profesori de la diverse școli din județ și reprezentanți ai societății civile. Pentru fiecare școală din județ se cunosc codul școlii, denumirea și numele directorului. Referitor la participanții la dezbatere se rețin numele, prenumele, ocupația și codul școlii, pentru reprezentanții societății civile codul școlii va fi NULL.

Afișați:

a) Numărul reprezentanților societății civile la dezbatere;

```
SELECT COUNT(*) FROM participanti WHERE codscoala IS NULL
```

b) Participanții elevi de la școala ...(denumirea școlii se introduce de la tastatură)

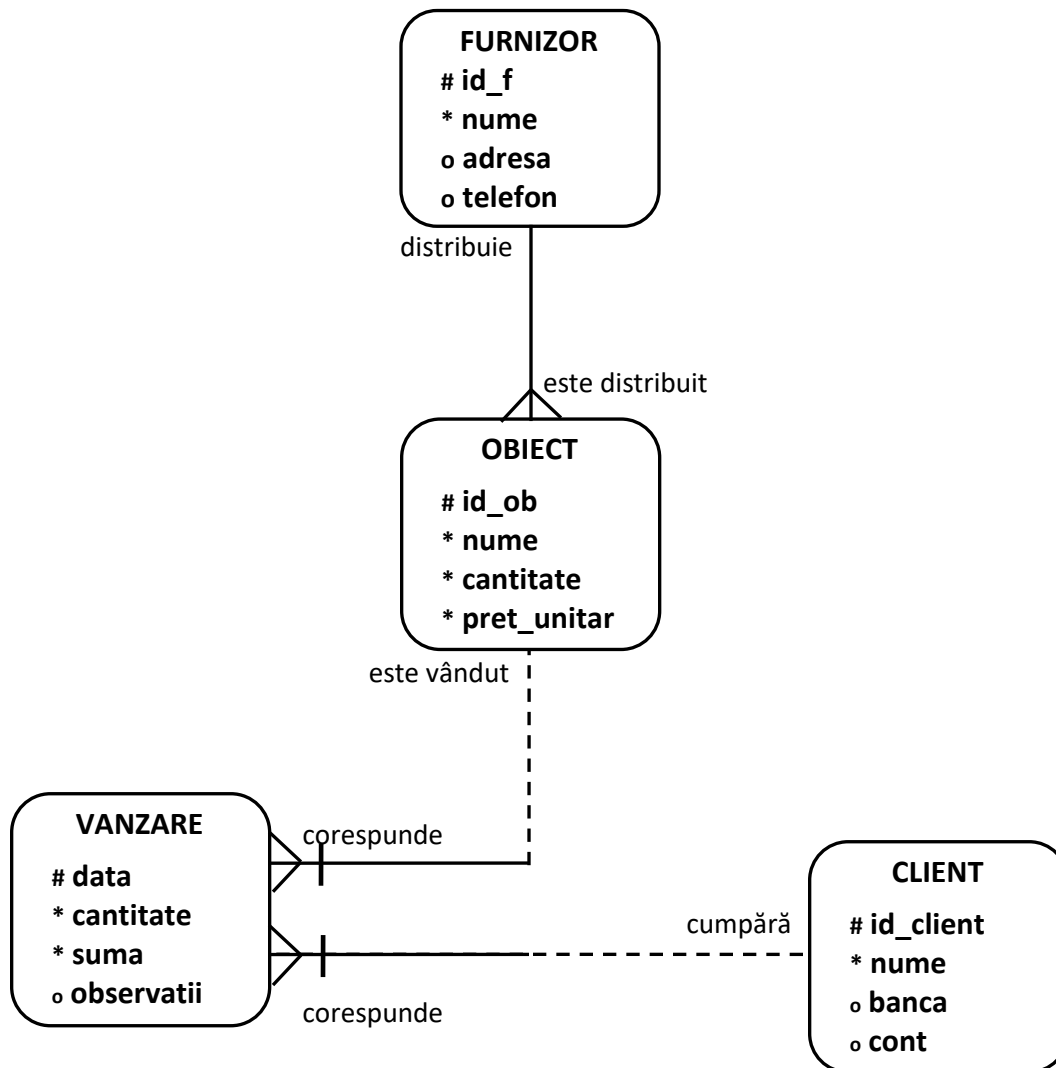
```
SELECT nume,prenume FROM participanti WHERE codscoala=:x AND ocupatia LIKE 'elev'
```

c) Directorul școlii/directorii școlilor cu cei mai mulți profesori participanți la dezbatere.

```
SELECT numedirector||' '||premedirector AS "Director" FROM scoli WHERE cod=(SELECT  
codscoala FROM participanti GROUP BY codscoala HAVING count(*)=(SELECT MAX(COUNT(*))  
FROM participanti GROUP BY ocupatia HAVING ocupatia='profesor')
```

Subiectul 21.

Realizați baza de date (tabelele) corespunzătoare și rezolvați cerințele pentru următoarea diagramă „entități-relații” (ERD):



RESTRICȚII:

- **Cantitate** (din Obiect) trebuie să conțină valori întregi pozitive;
- **Suma** (din Vanzare) trebuie să conțină valori pozitive cu 2 zecimale;

CERINȚE:

- Introduceți date în tabele (cel puțin câte 4 înregistrări);
- Afișați numele clienților care au cumpărat în luna decembrie a anului curent obiecte distribuite de furnizorul x (citit de la tastatură);

- Afișați numele obiectelor în ordine descrescătoare după cantitatea totală vândută.

CREATE TABLE Furnizori(id_f VARCHAR2(5) PRIMARY KEY, nume VARCHAR2(15) NOT NULL, adresa VARCHAR2(20), telefon VARCHAR2(10))

*Considerăm conținutul tabelului **Furnizori**:*

ID_F	NUME	ADRESA	TELEFON
f1	Adrian	B-dul I. Creanga 5	0102030405
F2	Iulian	Str. M. Eminescu, 12	0120340567
F3	Corina	Str. Negru Voda, 3	0234501678

CREATE TABLE Clienti(id_client VARCHAR2(5) PRIMARY KEY, nume VARCHAR2(15) NOT NULL, banca VARCHAR2(15), cont VARCHAR2(15))

*Considerăm conținutul tabelului **Clienti**:*

ID_CLIENT	NUME	BANCA	CONT
C1	Rares	BCR	B0011C12R12
C2	Ana-Maria	BRD	B0011R22D33
C3	Octavian	BCR	B0011C22R81

CREATE TABLE Obiecte(id_obiect VARCHAR2(5) PRIMARY KEY, nume VARCHAR2(15) NOT NULL, cantitate NUMBER(3) NOT NULL, pret_unitar NUMBER(5,2) NOT NULL, Id_f VARCHAR2(5) NOT NULL, FOREIGN KEY (Id_f) REFERENCES Furnizori(Id_f))

*Considerăm conținutul tabelului **Obiecte**:*

ID_OBIECT	NUME	CANTITATE	PRET_UNITAR	ID_F
Ob1	Birou	10	27,5	f1
Ob2	Scaun	15	42,5	f1
Ob3	Imprimanta	2	752,5	F2
Ob4	Tabla	2	325,25	F3

CREATE TABLE Vanzari(Data DATE, Cantitate NUMBER(3) NOT NULL, Suma NUMBER(7,2) NOT NULL, Observatii VARCHAR2(10), Id_client REFERENCES Clienti(Id_client), Id_obiect REFERENCES Obiecte(Id_obiect), PRIMARY KEY(Data, Id_client, Id_obiect))

*Considerăm conținutul tabelului **Vanzari**:*

DATA	CANTITATE	SUMA	OBSERVATII	ID_CLIENT	ID_OBIECT
15-11-2011	1	275,35	Birou	C2	Ob1
14-12-2011	1	325,25	Tabla	C1	Ob4
12-11-2011	2	90,5	Scaune	C1	Ob2
02-12-2011	1	752,75	imprimanta	C1	Ob3
07-12-2011	1	42,25	Scaun	C3	Ob2

- Numele clienților care au cumpărat în luna decembrie a anului curent obiecte distribuite de furnizorul x (citit de la tastatură):

```
SELECT Nume FROM Clienti WHERE Id_client IN
(SELECT Id_client FROM Vanzari WHERE TO_CHAR(Data,'mm-yyyy')='12-2011'
AND
Id_obiect IN(SELECT Id_obiect FROM Obiecte WHERE Id_f IN
(SELECT Id_f FROM Furnizori WHERE UPPER(Nume)= UPPER(:x))))
```

Nume client
Octavian

Pentru variabila :x s-a citit valoarea 'Adrian'.

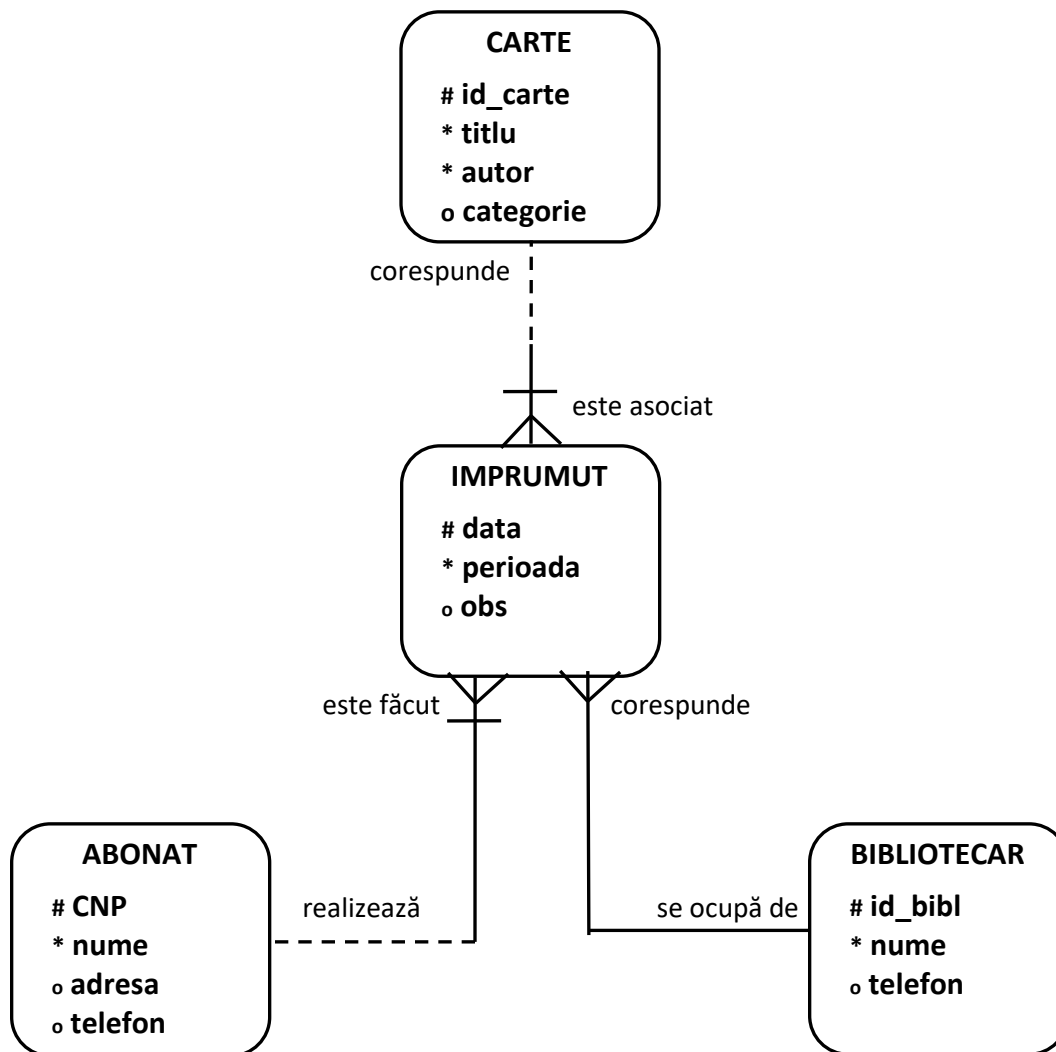
- Numele, cantitatea și prețul unitar al obiectelor grupate după numele furnizorului:

```
SELECT Nume "Nume obiect", Cantitate "Cantitate (buc.)",
Pret_unitar "Pret pe bucata", (SELECT Nume FROM Furnizori b WHERE
a.Id_f=b.Id_f) "Furnizor" FROM Obiecte a ORDER BY Id_f
```

Nume obiect	Cantitate (buc.)	Pret pe bucata	Furnizor
Birou	10	27,5	Adrian
Scaun	15	42,5	Adrian
Imprimanta	2	752,5	Iulian
Tabla	2	325,25	Corina

Subiectul 22.

Realizați baza de date (tabelele) corespunzătoare și rezolvați cerințele pentru următoarea diagramă „entități-relații” (ERD):



RESTRICȚII:

- **Categorie** (din Carte) poate fi: Roman, Aventuri, Beletristică sau Manual;
- **Perioada** (din Imprumut) reprezintă numărul de zile (maxim 21);

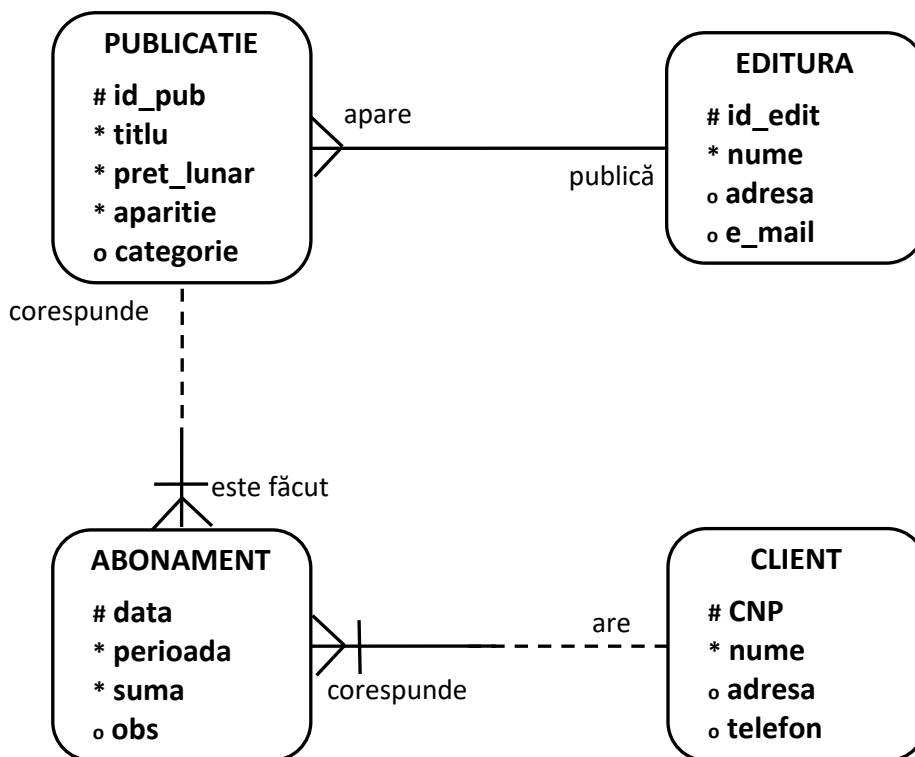
CERINȚE:

- Introduceți date în tabele (cel puțin câte 4 înregistrări);
- Afișați numele abonaților care ar trebui să înapoieze în luna curentă cărțile pe care le-au împrumutat de la bibliotecarul x (citit de la tastatură);

- Afișați titlul cărților în ordine descrescătoare după numărul total de împrumuturi.

Subiectul 23.

Realizați baza de date (tabelele) corespunzătoare și rezolvați cerințele pentru următoarea diagramă „entități-relații” (ERD):



RESTRICȚII:

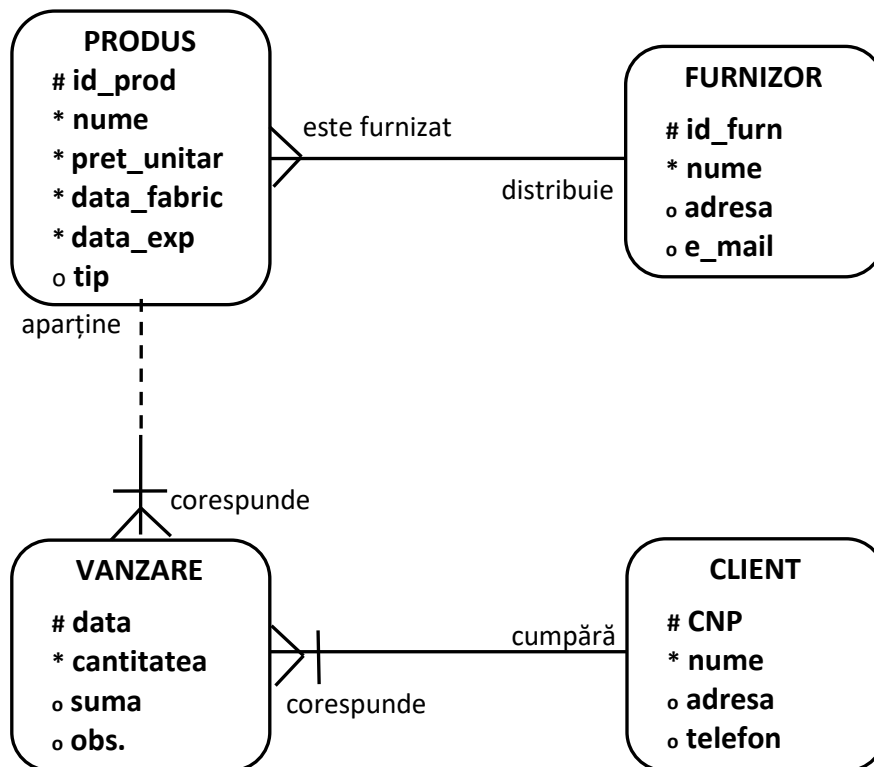
- **Categorie** (din Publicatie) poate fi: Ziar, Revista sau Almanah;
- **Aparitie** (din Publicatie) poate fi: Zilnica, Lunara, Bilunara sau Anuala;
- **Perioada** (din Abonament) reprezintă numărul de luni (maxim 12);

CERINȚE:

- Introduceți date în tabele (cel puțin câte 4 înregistrări);
- Afișați titlul publicațiilor care au apariție bilunară, pentru care clientul cu numele x (citit de la tastatură) are abonamente care se încheie în luna Mai a anului curent;
- Afișați numele editurilor ordonate descrescător după numărul total de abonamente făcute publicațiilor lor.

Subiectul 24.

Realizați baza de date (tabelele) corespunzătoare și rezolvați cerințele pentru următoarea diagramă „entități-relații” (ERD):



RESTRICȚII:

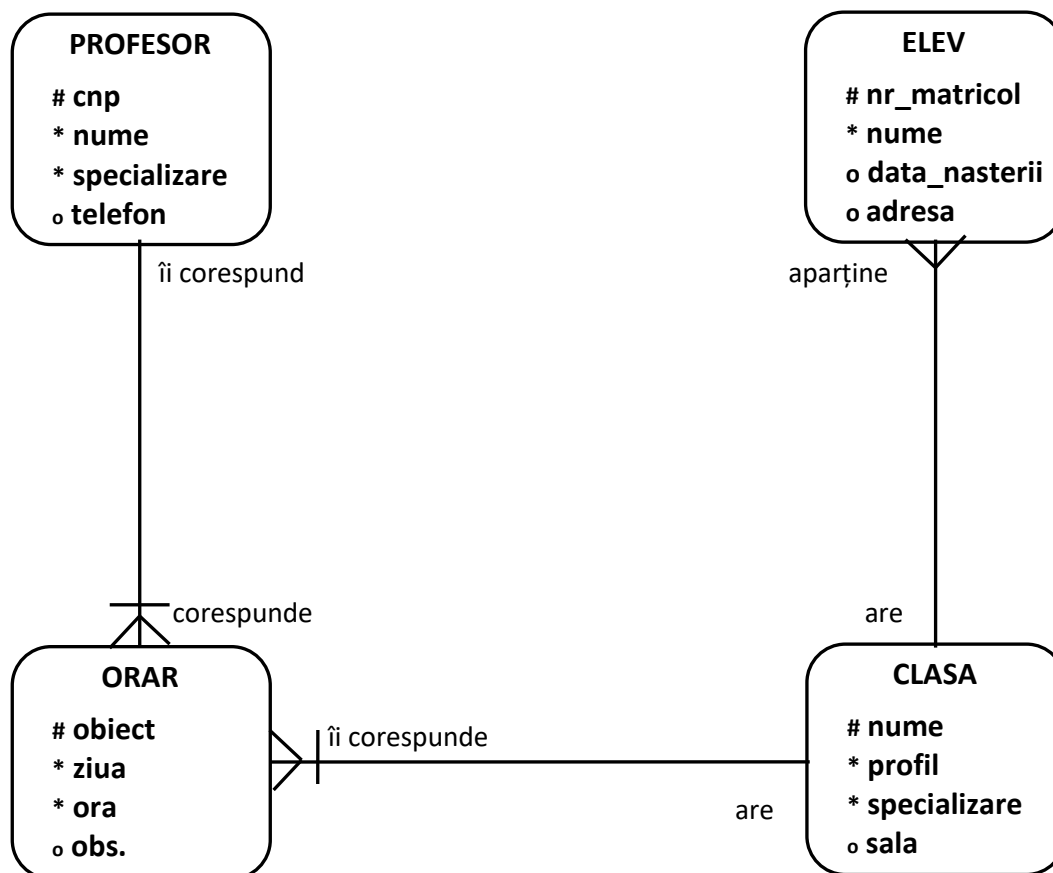
- **Tip** (din Produs) poate fi: Fructe, Legume sau Lactate;
- **Pret_unitar** (din Produs) este o valoare pozitivă cu 2 cifre zecimale;
- **Data_exp** (din Produs) reprezintă o dată ulterioară datei curente;

CERINȚE:

- Introduceți date în tabele (cel puțin câte 4 înregistrări);
- Afișați numele clienților care au cumpărat produse ce expiră în luna mai anul viitor, al căror furnizor are numele x (citit de la tastatură);
- Afișați numele furnizorilor de produse cu cea mai mare sumă totală a vânzărilor, ordonate descrescător după data fabricației.

Subiectul 25.

Realizați baza de date (tabelele) corespunzătoare și rezolvați cerințele pentru următoarea diagramă „entități-relații” (ERD):



RESTRICȚII:

- **Ziua**(din Orar) poate fi: Luni, Marți, Miercuri, Joi sau Vineri;
- **Nr_elevi** (din Clasa) poate fi un număr natural din intervalul [20,35];

CERINȚE:

- Introduceți date în tabele (cel puțin câte 4 înregistrări);
- Afișați numele și obiectele predate pentru profesorii predau în zilele de Marți la clasa în care se află elevul cu numele x (citit de la tastatură);

- Afișați numele claselor ordonate descrescător după numărul total de elevi, grupate după profil.

CREATE TABLE Profesori (cnp VARCHAR2(13) PRIMARY KEY, nume VARCHAR2(10), specializare VARCHAR2(10), telefon VARCHAR2(10))

Considerăm conținutul tabelului Profesori:

CNP	NUME	SPECIALIZARE	TELEFON
222	tomescu	mate	6789012345
333	crangus	romana	3456789012
111	merlan	info	1234567890

CREATE TABLE Clase(nume varchar2(5) PRIMARY KEY, profil VARCHAR2(10), specializare VARCHAR2(10), sala NUMBER(3))

Considerăm conținutul tabelului Clase:

NUME	PROFIL	SPECIALIZARE	SALA
10G	servicii	turism	8
9A	real	mate-info	9
10A	real	mate-info	16

CREATE TABLE Elevi(nr_matricol NUMBER(3) PRIMARY KEY, nume VARCHAR2(10), data_nasterii DATE, adresa VARCHAR2(20), clasa REFERENCES clase(nume))

Considerăm conținutul tabelului Elevi:

NR_MATRICOL	NUME	DATA_NASTERII	ADRESA	CLASA
100	nicoleta	19-02-1995	rozelor, nr.15	10G
10	maria	23-07-1996	cuza, nr.2	9A
11	alex	03-11-1996	muncii, nr.22	9A
50	ana	30-08-1995	radulescu, nr.2	10A
101	radu	09-04-1995	vladimirescu, nr.12	10G
51	george	13-05-1995	stirbei voda, nr.2	10A

CREATE TABLE Orar(obiect VARCHAR2(10), profesor VARCHAR2(13) REFERENCES profesori(cnp), clasa VARCHAR2(5) REFERENCES clase(nume), ziua VARCHAR2(10), ora NUMBER(5,2), obs VARCHAR2(10), PRIMARY KEY (obiect, profesor, clasa))

Considerăm conținutul tabelului Orar:

OBIECT	PROFESOR	CLASA	ZIUA	ORA	OBS
TIC	111	10G	MARTI	13	dm
TIC	111	10A	LUNI	16	dm
TIC	111	9A	joi	18	dm
mate	222	10G	MARTI	15	dm
romana	333	10G	MARTI	15	dm
romana	333	10A	MARTI	15	dm
romana	333	9A	JOI	14	dm
mate	222	10A	MARTI	10	AAA
mate	222	9A	joi	14	dm

- Numele și obiectele predate pentru profesorii care predau în zilele de Marți la clasa în care se află elevul cu numele x (citit de la tastatură):

```
SELECT nume, "Nume profesor", (SELECT obiect FROM orar a WHERE
b.cnp=a.profesor AND UPPER(ziua)='MARTI' AND clasa = (SELECT nume FROM
clase WHERE nume =
(SELECT clasa FROM elevi WHERE UPPER(nume) = UPPER(:x) ))) "Obiect predat"
FROM profesori b
```

Nume profesor	Obiect predat
tomescu	mate
crangus	romana
merlan	-

Pentru variabila :x s-a citit valoarea 'ana'.

Observăm că sunt afișați toți profesorii, indiferent dacă au sau nu ore în ziua de marți la clasa elevului :x. Acest lucru se întâmplă pentru că nu este pus niciun filtru pentru liniile tabelului Profesori.

Pentru a se afișa doar profesorii care îndeplinesc condițiile date, se poate folosi comanda:

```
SELECT nume "Nume profesor", (SELECT obiect FROM orar a WHERE
b.cnp=a.profesor AND UPPER(ziua)='MARTI' AND clasa = (SELECT nume FROM
clase WHERE nume =
(SELECT clasa FROM elevi WHERE UPPER(nume) = UPPER(:x) ))) "Obiect predat"
FROM profesori b WHERE
(SELECT obiect FROM orar a WHERE b.cnp=a.profesor AND UPPER(ziua)='MARTI'
AND clasa = (SELECT nume FROM clase WHERE nume = (SELECT clasa FROM elevi
WHERE UPPER(nume) = UPPER(:x)))) IS NOT NULL
```

Nume profesor	Obiect predat
tomescu	mate
crangus	romana

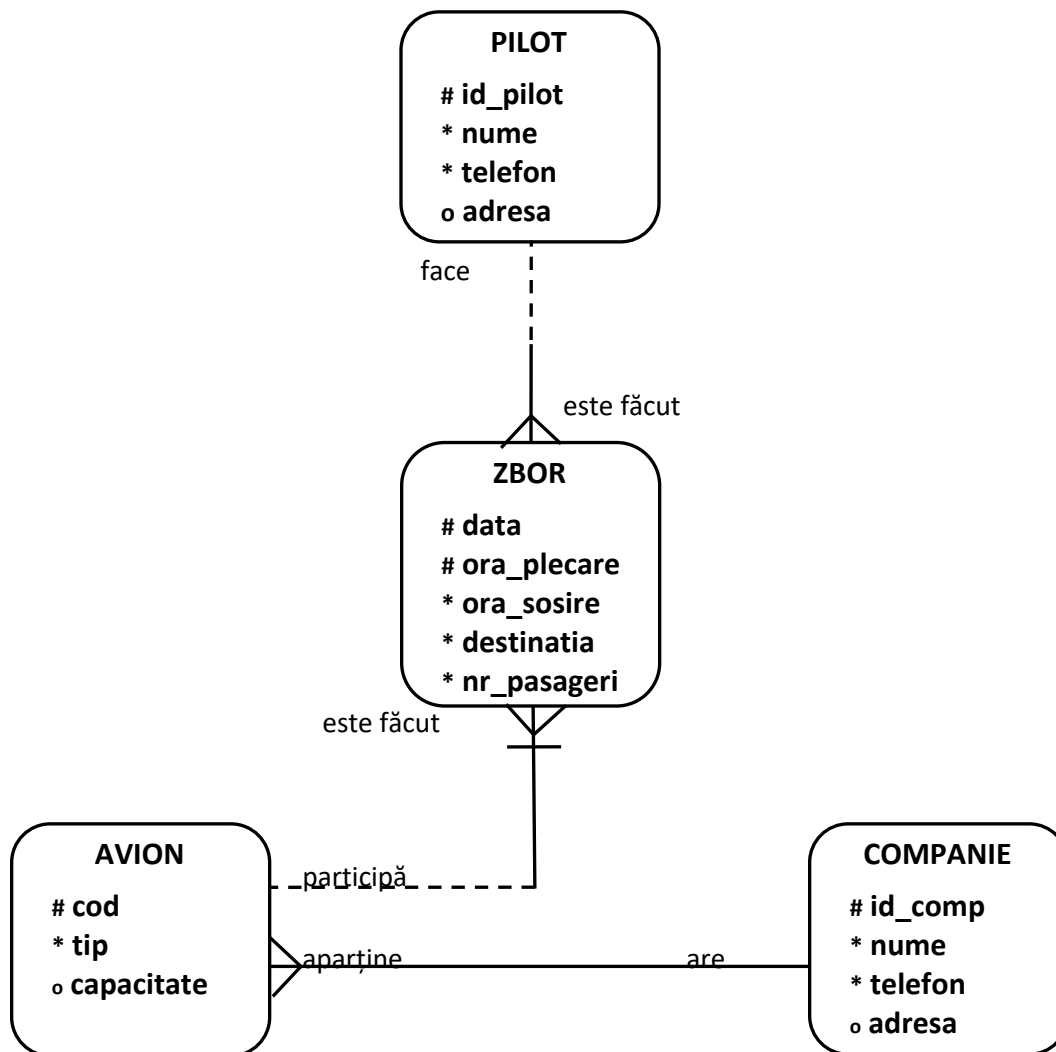
- Numele claselor și numărul total de elevi pentru fiecare profil:

SELECT nume "Clasa", (**SELECT** count(nr_matricol) **FROM** Elevi a **WHERE** b.nume=a.clasa) "Numar elevi", profil "Profil" **FROM** clase b **ORDER BY** profil

Clasa	Numar elevi	Profil
9A	2	real
10A	2	real
10G	2	servicii

Subiectul 26.

Realizați baza de date (tabelele) corespunzătoare și rezolvați cerințele pentru următoarea diagramă „entități-relații” (ERD):



RESTRICȚII:

- **Tip** (din Avion) poate fi: Airbus A380, Boeing B737 sau Wing 767;
- **Nr_pasageri** (din Zbor) este o valoare pozitivă mai mică decât 500);

CERINȚE:

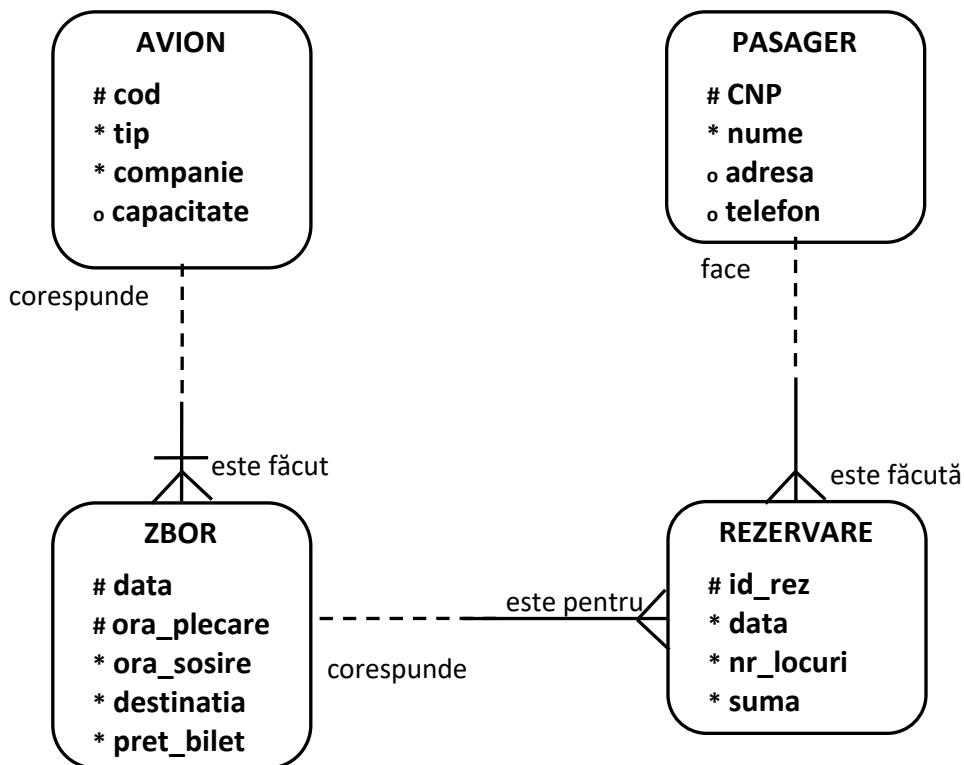
- Introduceți date în tabele (cel puțin câte 4 înregistrări);

- Afișați numele piloților care zboară în luna curentă, la ora 8 dimineața cu avioane de la compania cu numele x (citit de la tastatură);

Afișați data, ora, numărul de pasageri și codul avionului pentru toate zborurile, ordonate descrescător după numărul de pasageri.

Subiectul 27.

Realizați baza de date (tabelele) corespunzătoare și rezolvați cerințele pentru următoarea diagramă „entități-relații” (ERD):



RESTRICȚII:

- **Companie** (din Avion) poate fi: Air France, Alitalia, Blue Air și Lufthansa;
- **Data** (din Rezervare) trebuie să fie mai mare decât data curentă;
- **Nr_locuri** (din Rezervare) este număr natural (maxim 5);

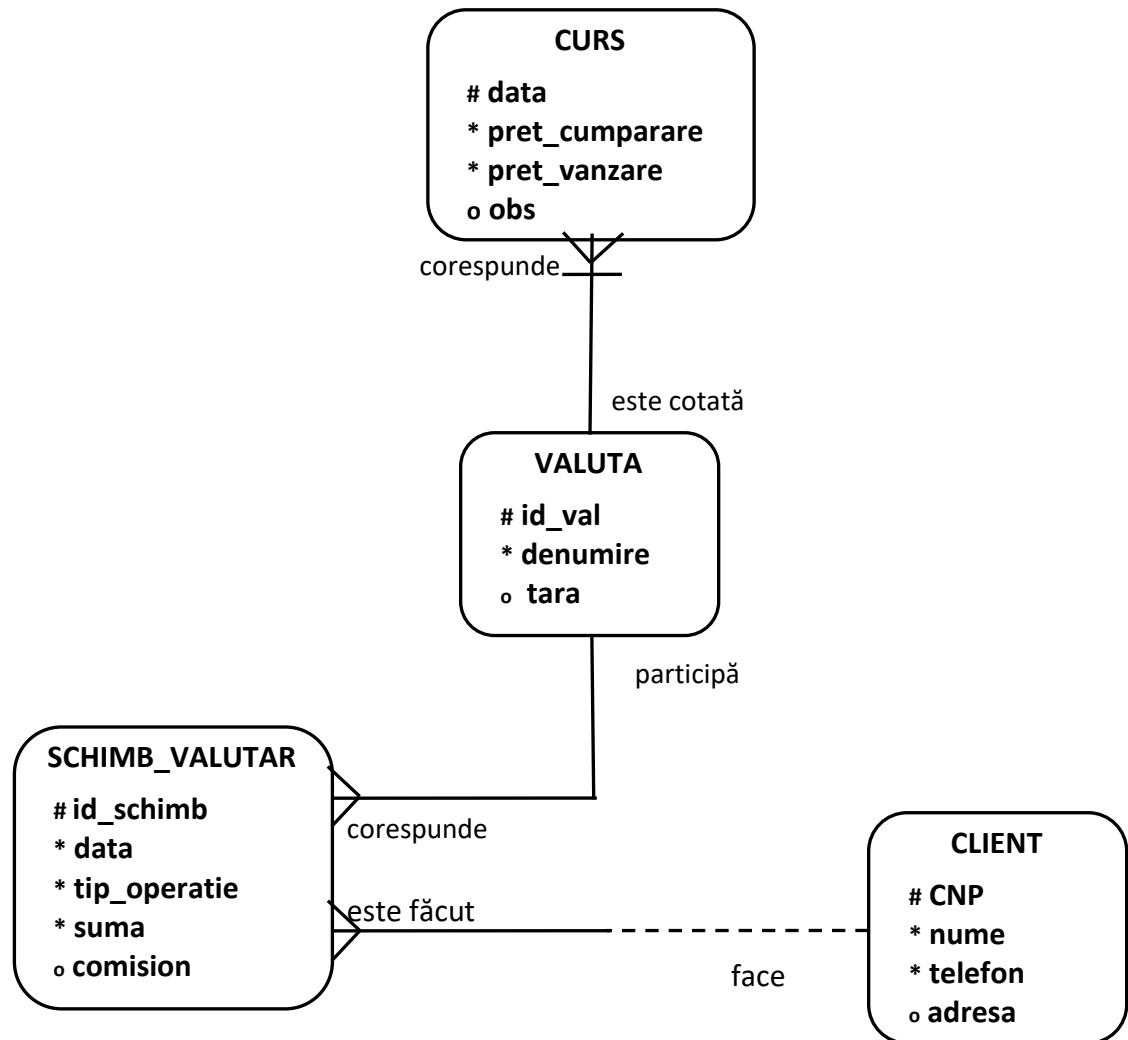
CERINȚE:

- Introduceți date în tabele (cel puțin câte 4 înregistrări);
- Afișați numele pasagerilor care și-au făcut rezervare la avioane de la o companie cu numele x (citit de la tastatură) care au ora plecării 12,30;

- Afișați codul și tipul avioanelor pentru toate zborurile, ordonate crescător după data și ora plecării.

Subiectul 28.

Realizați baza de date (tabelele) corespunzătoare și rezolvați cerințele pentru următoarea diagramă „entități-relații” (ERD):



RESTRICȚII:

- **Tip_operatie** (din Schimb_valutar) poate fi Vanzare sau Cumparare;
- **Comision** (din Schimb_valutar) este o valoare pozitivă mai mică decât 1,5;

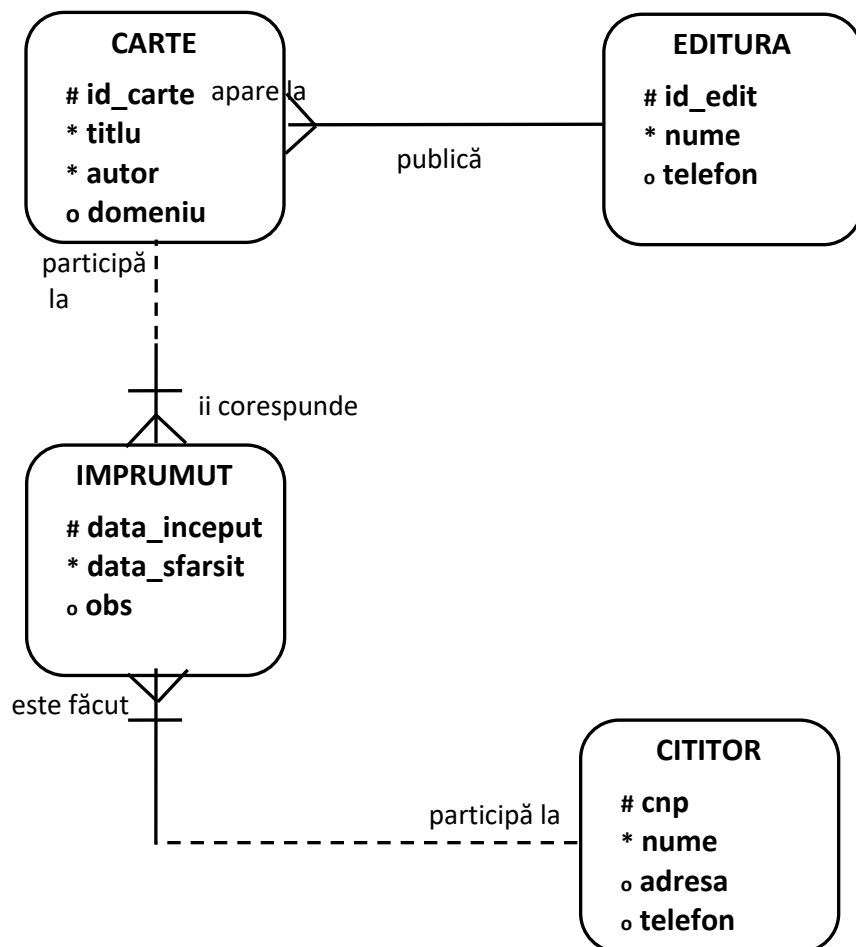
CERINȚE:

- Introduceți date în tabele (cel puțin câte 4 înregistrări);

- Afișați numele clienților care au cumpărat valuta cu numele x (citit de la tastatură) de sumă totală maximă;
- Afișați prețul de cumpărare și prețul de vânzare din data curentă pentru toate valutele, ordonate crescător după numele valutei.

Subiectul 29.

Realizați baza de date (tabelele) corespunzătoare și rezolvați cerințele pentru următoarea diagramă „entități-relații” (ERD):



RESTRICȚII:

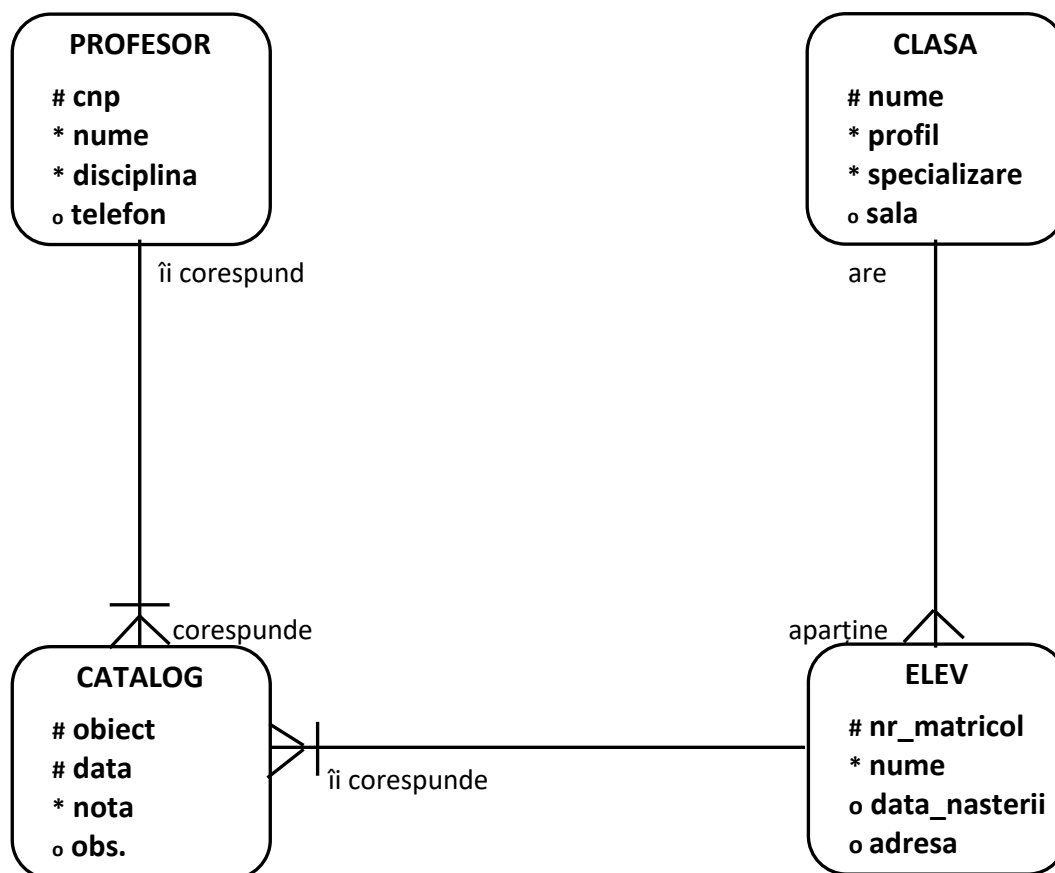
- **Domeniu** (din Carte) poate fi: SF, Stiinta, Filosofie sau Literatura;
- **Data_sfarsit** (din Imprumut) nu poate depăși data curentă cu mai mult de 14 zile;

CERINȚE:

- Introduceți date în tabele (cel puțin câte 4 înregistrări);
- Afișați numărul de cărți împrumutate de cititorul cu numele x (citit de la tastatură) în perioada 1 mai 2011 – 1 decembrie 2011;
- Afișați titlul cărților grupate după editură.

Subiectul 30.

Realizați baza de date (tabelele) corespunzătoare și rezolvați cerințele pentru următoarea diagramă „entități-relații” (ERD):



RESTRICȚII:

- **Nota** (din Catalog) poate fi: 1, 2, ..., 10;
- **Profil** (din Clasa) poate fi Real, Servicii sau Protecția mediului.

CERINȚE:

- Introduceți date în tabele (cel puțin câte 4 înregistrări);
- Afișați media la matematică pentru elevul cu numele x (citit de la tastatură);
- Afișați numele profesorilor care predau la fiecare clasă.