

# Proiect Baze de date



Nume: Mita

Prenume: Alexandra

Grupa: 5.2

Nr: Proiect 7

## Enunțul problemei:

Se considera o societate de creditare, pentru persoane fizice si juridice. Datele sunt structurate in urmatoarele tabele:

- **clienti\_fizice** avand urmatoarele campuri: ID, nume, prenume, telefon, CNP, adresa. Tabela cuprinde atat clientii persoane fizice cat si reprezentantii firmelor care sunt stocati in tabela **clienti\_juridice**.
- **clienti\_juridice** avand urmatoarele campuri: ID, nume, telefon, CUI, adresa, ID\_reprez. O firma poate avea un singur reprezentant.
- **credite** avand urmatoarele campuri: IDcredit, IDclient, tip\_client, suma, perioada, comision, dobanda, achitat. Un client poate avea mai multe credite pe numele sau.
- **plati** avand urmatoarele campuri: IDplata, IDcredit, data, suma. Pentru un credit se pot face oricate plati.
- **garantii** avand urmatoarele campuri: IDgarantie, IDcredit, descriere, suma, tip. Un credit poate avea mai multe garantii. O garantie nu poate fi utilizata la mai multe credite.

## Cerinte:

1. Sa se scrie comenzile de creare a tabelelor, impunand toate constrangerile care sunt considerate utile din punct de vedere functional si al integritatii datelor.
2. Sa se scrie comenzile de populare a tabelelor cu informatii (minim 5 articole in fiecare tabela).
3. Sa se afiseze totalul de bani aflati pe piata, sub forma:  
Clienti Plasament  
Persoane fizice xxxxxxxxxx  
Persoane juridice xxxxxxxxxx  
Total xxxxxxxxxx
4. Sa se afiseze resursele financiare disponibile ale societatii (banii care pot fi oferiti spre creditare clientilor) stiind ca initial societatea avea un capital de 1.000.000.
5. Stiind ca reprezentantii unei firme sunt stocati in tabela **clienti\_fizice**, indiferent daca au credite pe numele lor sau nu, legatura fiind asigurata de coloana ID\_reprez, sa se scrie interogarea care afiseaza o lista cu toti clientii persoane fizice care au credite atat pe numele lor, cat si pe firmele a caror reprezentanti sunt.
6. Sa se scrie interogarea care afiseaza toate firmele care au credite, ordonat dupa reprezentant. Afisarea va include: firma, CUI, nume reprezentant, prenume reprezentant, suma creditata, suma achitata.
7. Stiind ca fiecare garantie este stocata in tabela garantii, si ca un credit poate avea mai multe garantii, sa se afiseze lista creditelor care au garantii, grupat pe numarul de garantii.
8. Stiind ca tipul garantiilor specificate in tabela **garantii** poate fi: mobil, imobil si cec, plecand de la aceasta tabela sa se creeze o tabela **garant\_comb** avand urmatoarele campuri: IDcredit, tip\_garantie, suma, unde tip garantie este *mobil*, daca toate garantiile creditului sunt mobile, *imobil*, daca toate garantiile creditului sunt imobile, *cec*, daca toate garantiile creditului sunt cecuri si *combinat* daca creditul are garantii

de diferite tipuri.

9. Considerand tabela de la punctul 8 creata, sa se scrie un trigger care la orice modificare a tabelii garantii, actualizeaza automat si tabela **garant\_comb**.

10. Sa se scrie un trigger care la orice modificare in tabela plati sa actualizeze coloana **achitat** din tabela credite.

## Rezolvare:

1. Întâi am creat tabelele cerute din enunț, impunând constrângerile ce asigură funcționarea aplicației în mod corect. Astfel, vom avea 5 tabele: **clienti\_fizice**, **clienti\_juridice**, **credite**, **plati**, **garantii**.

```
CREATE table clienti_fizice(  
ID int primary key,  
nume varchar(30) NOT NULL,  
prenume varchar(30) NOT NULL,  
telefon number(10) NOT NULL,  
CNP number(13) NOT NULL,  
adresa varchar(30)NOT NULL  
);
```

```
SQL> CREATE table clienti_fizice(  
2  ID int primary key,  
3  nume varchar(30) NOT NULL,  
4  prenume varchar(30) NOT NULL,  
5  telefon number(10) NOT NULL,  
6  CNP number(13) NOT NULL,  
7  adresa varchar(30)NOT NULL  
8  );
```

Table created.

```
CREATE TABLE clienti_juridice(  
ID int PRIMARY KEY,  
nume varchar(30) NOT NULL,  
telefon number(10) NOT NULL,  
CUI int NOT NULL,  
adresa varchar(30) NOT NULL,  
ID_reprez int REFERENCES clienti_fizice(ID)  
);
```

```
SQL> CREATE TABLE clienti_juridice(
  2     ID int PRIMARY KEY,
  3     nume varchar(30) NOT NULL,
  4     telefon number(10) NOT NULL,
  5     CUI int NOT NULL,
  6     adresa varchar(30) NOT NULL,
  7     ID_reprez int REFERENCES clienti_fizice(ID)
  8 );
```

Table created.

```
CREATE TABLE credite(
IDcredit int PRIMARY KEY,
IDclient int REFERENCES clienti_fizice(ID) ON DELETE CASCADE,
tip_client varchar(15) NOT NULL,
suma int,
perioada number(2) CHECK (perioada BETWEEN 1 AND 31),
comision number(7,2),
dobanda number(7,2),
achitat int
);
```

```
SQL> CREATE TABLE credite(
  2 IDcredit int PRIMARY KEY,
  3 IDclient int REFERENCES clienti_fizice(ID) ON DELETE CASCADE,
  4 tip_client varchar(15) NOT NULL,
  5 suma int,
  6 perioada number(2) CHECK (perioada BETWEEN 1 AND 31),
  7 comision number(7,2),
  8 dobanda number(7,2),
  9 achitat int
 10 );
```

Table created.

```
CREATE table plati(
IDplata int,
IDcredit int references credite(IDcredit) on delete cascade,
data date,
suma int
);
```

```
SQL> CREATE table plati(
  2 IDplata int,
  3 IDcredit int references credite(IDcredit) on delete cascade,
  4 data date,
  5 suma int
  6 );
```

Table created.

```
CREATE table garantii(
```

IDgarantie int PRIMARY KEY,  
IDcredit int references credite(IDcredit) on delete cascade,  
descriere varchar(30),  
suma int,  
tip varchar(10) NOT NULL  
);

```
SQL> CREATE table garantii(  
2 IDgarantie int PRIMARY KEY,  
3 IDcredit int references credite(IDcredit) on delete cascade,  
4 descriere varchar(30),  
5 suma int,  
6 tip varchar(10) NOT NULL  
7 );
```

Table created.

## 2. Popularea tabelor cu valori cu minim 5 articole

```
INSERT into clienti_fizice  
values(1,'Apăvăloaei','Mihai',0734567891,5031126027951,'Strada Traian Nerva');  
INSERT into clienti_fizice  
values(2,'Vrăjoiu','Mara',0777452131,6030509025661,'Strada Aurel Vlaicu');  
INSERT into clienti_fizice values(3,'Uliu' ,' Sonia',0700100100,6030908026719,'Strada  
Decebal');  
INSERT into clienti_fizice  
values(4,'Luțu','Adelina',0798765432,6031220409683,'Strada Neptun ');  
INSERT into clienti_fizice  
values(5,'Pătrar','Denisa',0721455689,6030811408188,'Strada Jupiter');  
INSERT into clienti_fizice  
values(6,'Piatră','Antonia',0721455689,6030824407294,'Strada Mihai Eminescu');
```

```
SQL> INSERT into clienti_fizice values(1,'Apăvăloaei','Mihai',0734567891,5031126027951,'Strada Traian Nerva');  
1 row created.  
SQL> INSERT into clienti_fizice values(2,'Vrăjoiu','Mara',0777452131,6030509025661,'Strada Aurel Vlaicu');  
1 row created.  
SQL> INSERT into clienti_fizice values(3,'Uliu' ,' Sonia',0700100100,6030908026719,'Strada Decebal');  
1 row created.  
SQL> INSERT into clienti_fizice values(4,'Luțu','Adelina',0798765432,6031220409683,'Strada Neptun ');  
1 row created.  
SQL> INSERT into clienti_fizice values(5,'Pătrar','Denisa',0721455689,6030811408188,'Strada Jupiter');  
1 row created.  
SQL> INSERT into clienti_fizice values(6,'Piatră','Antonia',0721455689,6030824407294,'Strada Mihai Eminescu');  
1 row created.
```

```
INSERT into clienti_juridice  
values(7,'Continental',0734567891,5031126027951,'Strada Traian Nerva',1 );  
INSERT into clienti_juridice values(8,'Google',0777452131,6030509025661,'Strada  
Aurel Vlaicu',2 );  
INSERT into clienti_juridice values(9,'Amazon',0700100100,6030908026719,'Strada  
Decebal',3 );
```

INSERT into clienti\_juridice values(10,'Hella',0798765432,6031220409683,'Strada Neptun',4 );

INSERT into clienti\_juridice values(11,'Dior',0721455689,6030811408188,'Strada Jupiter',5 );

INSERT into clienti\_juridice values(12,'Tesla',0721455689,6030824407294,'Strada Mihai Eminescu',6 );

```
SQL> INSERT into clienti_juridice values(7,'Continental',0734567891,5031126027951,'Strada Traian Nerva',1 );
1 row created.

SQL> INSERT into clienti_juridice values(8,'Google',0777452131,6030509025661,'Strada Aurel Vlaicu',2 );
1 row created.

SQL> INSERT into clienti_juridice values(9,'Amazon',0700100100,6030908026719,'Strada Decebal',3 );
1 row created.

SQL> INSERT into clienti_juridice values(10,'Hella',0798765432,6031220409683,'Strada Neptun',4 );
1 row created.

SQL> INSERT into clienti_juridice values(11,'Dior',0721455689,6030811408188,'Strada Jupiter',5 );
1 row created.

SQL> INSERT into clienti_juridice values(12,'Tesla',0721455689,6030824407294,'Strada Mihai Eminescu',6 );
1 row created.
```

INSERT into credite values(111,1,'persoana',202020,8,0.85,8,2020);

INSERT into credite values(222,2,'persoana',10000,2,1.25,9.5,6000);

INSERT into credite values(333,3,'organizatie',56000,4,5.5,18,6600);

INSERT into credite values(444,4,'firma',225000,11,9,15.5,1000);

INSERT into credite values(555,1,'firma',300000,20,4.5,16,25000);

```
SQL> INSERT into credite values(111,1,'persoana',202020,8,0.85,8,2020);
1 row created.

SQL> INSERT into credite values(222,2,'persoana',10000,2,1.25,9.5,6000);
1 row created.

SQL> INSERT into credite values(333,3,'organizatie',56000,4,5.5,18,6600);
1 row created.

SQL> INSERT into credite values(444,4,'firma',225000,11,9,15.5,1000);
1 row created.

SQL> INSERT into credite values(555,1,'firma',300000,20,4.5,16,25000);
1 row created.
```

INSERT into plati values(211,111,'19-mai-2012',1000);

INSERT into plati values(222,222,'22-septembrie-2018',2000);

INSERT into plati values(233,555,'10-iunie-2005',500);

INSERT into plati values(244,111,'30-iulie-2020',800);

INSERT into plati values(255,444,'23-august-2002',3250);

```
SQL> INSERT into plati values(222,222,'22-septembrie-2018',2000);
1 row created.
SQL> INSERT into plati values(233,555,'10-iunie-2005',500);
1 row created.
SQL> INSERT into plati values(244,111,'30-iulie-2020',800);
1 row created.
SQL> INSERT into plati values(255,444,'23-august-2002',3250);
1 row created.
```

```
INSERT into garantii values(311,111,'garantie scurta durata',5000,'mobil');
INSERT into garantii values(322,222,'garantie lungadurata',10000,'imobil');
INSERT into garantii values(333,333,'garantie pe 2 ani',7000,'cec');
INSERT into garantii values(344,444,'garantie ce expira',45000,'mobil');
INSERT into garantii values(355,555,'garantie speciala',1500,'imobil');
```

```
SQL> INSERT into garantii values(311,111,'garantie scurta durata',5000,'mobil');
1 row created.
SQL> INSERT into garantii values(322,222,'garantie lungadurata',10000,'imobil');
1 row created.
SQL> INSERT into garantii values(333,333,'garantie pe 2 ani',7000,'cec');
1 row created.
SQL> INSERT into garantii values(344,444,'garantie ce expira',45000,'mobil');
1 row created.
SQL> INSERT into garantii values(355,555,'garantie speciala',1500,'imobil');
1 row created.
```

3. Facem afişarea întâi ca şi suma de bani obţinută în funcţie de tipul clientului, apoi suma totală a creditelor

```
SELECT tip_client AS clienti, sum(suma) AS plasament
FROM credite
GROUP BY tip_client;
```

```
SQL> SELECT tip_client AS clienti, sum(suma) AS plasament
2 FROM credite
3 GROUP BY tip_client;
```

CLIENTI	PLASAMENT
persoana	212020
firma	525000
organizatie	56000

```
SELECT sum(suma) AS total FROM credite;
```

```
SQL> select sum(suma) as total from credite;
```

TOTAL
793020

4. Pentru a afișa resursele disponibile va trebui să scădem din întregul capital oferit de cerința problemei suma deja cheltuită

```
SELECT 1000000-sum(suma) AS resurse FROM credite;
```

```
SQL> select 1000000-sum(suma) as resurse from credite;
```

RESURSE
206980

5. Vom folosi o interogare care va selecta toți clienții persoane fizice împreună cu cei care sunt reprezentanți ai unei firme ce dețin credite. Astfel, se selectează toate ID-urile reprezentanților de firme din tabela clienti\_juridice, apoi selectează toți clienții persoane fizice care au credite. Aceștia sunt selectați din tabela credite, unde ID-ul clientului coincide cu ID-ul unui client persoană fizică. Obținem următorul lucru:

```
SQL> SELECT DISTINCT cf.num, cf.prenume
2 FROM clienti_fizice cf
3 JOIN credite c ON cf.ID = c.IDclient
4 WHERE cf.ID IN (
5     SELECT DISTINCT ID_reprez
6     FROM clienti_juridice
7 ) AND c.tip_client = 'persoana';
```

NUME	PRENUME
Apăvăloaei	Mihai
Vrăjoiu	Mara

6. Ne folosim de index

```
CREATE INDEX ex6 ON clienti_juridice(ID_reprez);
```

Interogarea selectează informațiile necesare, grupând rezultatele după numele și prenumele reprezentantului și după numele firmei.

```
SELECT c_juridic.num AS firma,
c_juridic.CUI,
c_fizic.num AS nume_reprezentant,
c_fizic.prenume AS prenume_reprezentant,
SUM(c.suma) AS suma_creditata,
```



```

COALESCE(SUM(p.suma), 0) AS suma_achitata
FROM clienti_juridice c_juridic
JOIN clienti_fizice c_fizic ON c_juridic.ID_reprez = c_fizic.ID
LEFT JOIN credite c ON c_juridic.ID = c.IDclient
LEFT JOIN plati p ON c.IDcredit = p.IDcredit
GROUP BY c_juridic.ume, c_juridic.CUI, c_fizic.ume, c_fizic.prenume
ORDER BY c_fizic.ume, c_fizic.prenume;

```

```

SQL> SELECT c_juridic.ume AS firma,
2         c_juridic.CUI,
3         c_fizic.ume AS ume_reprezentant,
4         c_fizic.prenume AS prenume_reprezentant,
5         SUM(credite.suma) AS suma_creditata,
6         COALESCE(SUM(plati.suma), 0) AS suma_achitata
7 FROM clienti_juridice c_juridic
8 JOIN clienti_fizice c_fizic ON c_juridic.ID_reprez = c_fizic.ID
9 LEFT JOIN credite ON c_juridic.ID = credite.IDclient
10 LEFT JOIN plati ON credite.IDcredit = plati.IDcredit
11 GROUP BY c_juridic.ume, c_juridic.CUI, c_fizic.ume, c_fizic.prenume
12 ORDER BY c_fizic.ume, c_fizic.prenume;

```

FIRMA	CUI	UME_REPREZENTANT	
PRENUME_REPREZENTANT	SUMA_CREDITATA	SUMA_ACHITATA	
Continental Mihai	5,0311E+12	Apăvăloaei	0
Hella Adelina	6,0312E+12	Lu?u	0
Dior Denisa	6,0308E+12	Pătrar	0

  

FIRMA	CUI	UME_REPREZENTANT	
PRENUME_REPREZENTANT	SUMA_CREDITATA	SUMA_ACHITATA	
Tesla Antonia	6,0308E+12	Piatră	0
Amazon Sonia	6,0309E+12	Uliu	0
Google Mara	6,0305E+12	Vrăjoiu	0

7.

```

CREATE VIEW ex7 AS (
SELECT
    COUNT(IDcredit) AS numar,
    IDcredit AS credit
FROM garantii
GROUP BY IDcredit
);

```

```

SELECT e.numar, e.credit, c.tip_client, c.suma, c.perioada, c.comision
FROM ex7 e
JOIN credite c ON e.credit = c.IDcredit;

```

În acest caz am creeat un view pentru a vizualiza condiția pusă la această cerință, numărând garanțiile care vor fi puse in view-ul creat.

```

SQL> CREATE VIEW ex7 AS (
2     SELECT
3         COUNT(IDcredit) AS numar,
4         IDcredit AS credit
5     FROM
6         garantii
7     GROUP BY
8         IDcredit
9 );

```

View created.

```

SQL>
SQL> SELECT
2     e.numar,
3     e.credit,
4     c.tip_client,
5     c.suma,
6     c.perioada,
7     c.comision
8 FROM
9     ex7 e
10 JOIN
11     credite c ON e.credit = c.IDcredit;

```

NUMAR	CREDIT	TIP_CLIENT	SUMA	PERIOADA	COMISION
1	111	persoana	202020	8	,85
1	222	persoana	10000	2	1,25
1	333	organizatie	56000	4	5,5
1	444	firma	225000	11	9
1	555	firma	300000	20	4,5

În acest mod putem numara cate garanții se află pe baza unui singur credit utilizându-se ID-ul specific.

```

SQL> SELECT cr.IDcredit, COUNT(*) AS numar_garantii
2 FROM garantii g
3 INNER JOIN credite cr ON g.IDcredit = cr.IDcredit
4 GROUP BY cr.IDcredit;

```

IDCREDIT	NUMAR_GARANTII
555	1
222	1
444	1
333	1
111	1

## 8. Trebuie creată tabela garant\_comb

```
INSERT INTO garant_comb (IDcredit, tip_garantie, suma)
SELECT g.IDcredit,
CASE
WHEN COUNT(DISTINCT g.tip) = 1 AND MAX(g.tip) = 'mobil' THEN 'mobil'
WHEN COUNT(DISTINCT g.tip) = 1 AND MAX(g.tip) = 'imobil' THEN 'imobil'
WHEN COUNT(DISTINCT g.tip) = 1 AND MAX(g.tip) = 'cec' THEN 'cec'
ELSE 'combinat'
END AS tip_garantie,
SUM(g.suma) AS suma
FROM garantii g
GROUP BY g.IDcredit;
```

```
SQL> INSERT INTO garant_comb (IDcredit, tip_garantie, suma)
2  SELECT
3      g.IDcredit,
4      CASE
5          WHEN COUNT(DISTINCT g.tip) = 1 AND MAX(g.tip) = 'mobil' THEN 'mobil'
6          WHEN COUNT(DISTINCT g.tip) = 1 AND MAX(g.tip) = 'imobil' THEN 'imobil'
7          WHEN COUNT(DISTINCT g.tip) = 1 AND MAX(g.tip) = 'cec' THEN 'cec'
8          ELSE 'combinat'
9      END AS tip_garantie,
10     SUM(g.suma) AS suma
11 FROM garantii g
12 GROUP BY g.IDcredit;

5 rows created.
```

```
SQL> select * from garant_comb
2  ;
```

IDCREDIT	TIP_GARANT	SUMA
555	imobil	1500
222	imobil	10000
444	mobil	45000
333	cec	7000
111	mobil	5000
555	imobil	1500
222	imobil	10000
444	mobil	45000
333	cec	7000
111	mobil	5000
555	imobil	1500

  

IDCREDIT	TIP_GARANT	SUMA
222	imobil	10000
444	mobil	45000
333	cec	7000
111	mobil	5000

```
15 rows selected.
```

```
SELECT IDcredit,
CASE
WHEN MIN(tip_garantie) = MAX(tip_garantie) THEN
```

```

CASE
WHEN MIN(tip_garantie) = 'mobil' THEN 'toate garantiile sunt mobile'
WHEN MIN(tip_garantie) = 'imobil' THEN 'toate garantiile sunt imobile'
WHEN MIN(tip_garantie) = 'cec' THEN 'toate garantiile sunt cecuri'
END
ELSE 'combinat'
END AS tip_garantie
FROM garant_comb
GROUP BY IDcredit;

```

Această interogare oferă o privire asupra tipului general de garanție asociat fiecărui credit și specifică dacă toate garanțiile asociate creditului sunt mobile, imobile, cecuri sau combinat.

```

SQL> SELECT
2     IDcredit,
3     CASE
4         WHEN MIN(tip_garantie) = MAX(tip_garantie) THEN
5             CASE
6                 WHEN MIN(tip_garantie) = 'mobil' THEN 'toate garantiile sunt mobile'
7                 WHEN MIN(tip_garantie) = 'imobil' THEN 'toate garantiile sunt imobile'
8                 WHEN MIN(tip_garantie) = 'cec' THEN 'toate garantiile sunt cecuri'
9             END
10        ELSE 'combinat'
11    END AS tip_garantie
12 FROM
13     garant_comb
14 GROUP BY
15     IDcredit;

```

IDCREDIT	TIP_GARANTIE
555	toate garantiile sunt imobile
222	toate garantiile sunt imobile
444	toate garantiile sunt mobile
333	toate garantiile sunt cecuri
111	toate garantiile sunt mobile

9.

```

CREATE OR REPLACE TRIGGER ex9
AFTER INSERT OR UPDATE ON garantii
FOR EACH ROW
BEGIN
    UPDATE garant_comb
    SET suma = suma + :new.suma
    WHERE idcredit = :new.idcredit;
END;
/

```

```

SQL> CREATE OR REPLACE TRIGGER ex9
2     AFTER INSERT OR UPDATE ON garantii
3     FOR EACH ROW
4     BEGIN
5         UPDATE garant_comb
6         SET suma = suma + :new.suma
7         WHERE idcredit = :new.idcredit;
8     END;
9     /

```

Trigger created.

Când se face o nouă inserare sau actualizare în tabela plati, coloana achitat din tabela "credite" va fi actualizată cu suma totală a plăților.

10.

```
CREATE OR REPLACE TRIGGER trig_plati
AFTER INSERT OR UPDATE ON plati
FOR EACH ROW
BEGIN
UPDATE credite
SET achitat = ( SELECT SUM(suma)
FROM plati WHERE plati.IDcredit = credite.IDcredit)
WHERE IDcredit = :new.IDcredit;
END;
/
```

```
SQL> CREATE OR REPLACE TRIGGER trig_plati
 2 AFTER INSERT OR UPDATE ON plati
 3 FOR EACH ROW
 4 BEGIN
 5 UPDATE credite
 6 SET achitat = ( SELECT SUM(suma)
 7 FROM plati WHERE plati.IDcredit = credite.IDcredit)
 8 WHERE IDcredit = :new.IDcredit;
 9 END;
10 /

Trigger created.
```

Acest trigger făcut va fi declanșat după fiecare operație de insert sau update în tabela plati. Scopul său este de a da update la coloana achitat pentru creditul asociat din tabela plăți.