

Gestiunea unei retele de socializare

- Vatau Lorena -
Grupa 133

STUDIU DE CAZ – EX. 1

Modelul real al unei baze de date "rețea de socializare" reflectă structura și relațiile dintre entitățile și obiectele prezente într-o platformă de socializare.

În cadrul acestei rețele de socializare, crearea bazei de date are scopul de a gestiona informațiile legate de rețea, fondator, adrese, utilizatori și angajați. Aceasta permite menținerea unei evidențe complete și organizate a acestor entități și a interacțiunilor dintre ele.

Baza de date are un design bine structurat și relațional, în care tabelele sunt conectate prin relații și constrângeri. Prin utilizarea acestui model, se asigură integritatea datelor și posibilitatea realizării unor interogări complexe și analize aprofundate.

De exemplu, tabela "Profil" este conectată la tabela "Utilizator" prin intermediul unei chei străine, ceea ce permite asocierea profilului unui utilizator. Totodată, tabela "Angajat" este conectată la tabela "Departament" prin intermediul unei chei străine, permițând asocierea angajaților cu departamentele în care lucrează.

Prin intermediul bazei de date, se pot realiza diverse operațiuni și analize. De exemplu, se pot extrage informații despre utilizatori și interacțiunile lor, se pot efectua căutări în funcție de anumite criterii sau se pot realiza rapoarte și statistici legate de activitatea utilizatorilor.

Astfel, baza de date reprezintă un instrument esențial în gestionarea și analizarea informațiilor din rețeaua de socializare, oferind o perspectivă detaliată asupra utilizatorilor, angajaților, fondatorilor, interacțiunilor lor etc.. Prin intermediul acestei baze de date, se poate asigura o funcționare eficientă și coerentă a rețelei de socializare, facilitând gestionarea și analiza datelor într-un mod optim.

REGULI DE FUNCTIONARE – EX2

Baza de date respecta anumite reguli de functionare. Acestea sunt:

- Un fondator a creat cel putin o retea, iar o retea nu poate sa aiba mai multi fondatori
- O retea poate sa aiba o singura adresa
- Adresa de infiintare se gaseste la o localitate
- Localitate are cel putin o adresa si are un judet
- Un judet are cel putin o localitate
- O recenzie se gaseste pe cel putin un profil, iar un profil are cel putin o recenzie
- O postare este postata pe cel putin un profil, iar un profil are cel putin o postare
- Fiecare profil este accesat de un singur utilizator, iar un utilizator poate sa detina un singur profil
- Un utilizator se gaseste pe mai multe retele sociale, iar o retea contine mai multi utilizatori
- Reteaua sociala este impartita pe mai multe departamente, iar departamentele sunt asociate unei singure retele
- Un department are cel putin un angajat
- Angajatul poate sa lucreze la un singur department
- Fiecare angajat are un singur job
- Un job este practicat de cel putin un angajat

ENTITATI – EX. 3

În această bază avem următoarele entități independente: FONDATOR, RETEA_SOCIALA, ADRESA_INFIINTARE, LOCALITATE, JUDET, DEPARTAMENT, ANGAJATI, JOB, UTILIZATOR, PROFIL, POSTARE, RECENZIE.

Entitățile dependente sunt: NOTIFICARE, CONT.

FONDATOR = persoana fizică care a avut ideea și a investit resursele necesare pentru a crea și menține rețeaua. Cheia primară este id_fondator.

RETEA_SOCIALA = platforma online pe care utilizatorii se înregistrează pentru a interacționa cu alți utilizatori și pentru a partaja conținut, cum ar fi postări. Cheia primară este id_retea.

ADRESA_INFIINTARE = adresa fizică a sediului principal care conține informații despre locul în care rețeaua de socializare a fost înființată. Cheia primară este id_adresa.

LOCALITATE = descrie diferitele locații geografice asociate cu rețeaua de socializare și adresa de înființare. Cheia primară este id_localitate.

JUDET = unitatea administrativă locală la nivel de județ asociată cu rețeaua de socializare. Cheia primară este id_judet.

DEPARTAMENT = diviziunile interne ale companiei responsabile de gestionarea și dezvoltarea rețelei de socializare. Cheia primară este id_departament.

ANGAJATI = persoanele fizice angajate în cadrul companiei care operează rețeaua de socializare. Cheia primară este id_angajati.

JOB = posturi sau roluri pe care angajații le pot ocupa în cadrul companiei responsabile de rețeaua de socializare. Cheia primară este id_job.

UTILIZATORI = persoanele înregistrate în rețeaua de socializare, care au conturi personale și interacționează cu alți utilizatori. Cheia primară este id_utilizatori.

PROFIL = conține informații detaliate despre profilurile utilizatorilor din rețeaua de socializare. Cheia primara este id_profil.

POSTARE = conținutul creat și partajat de către utilizatori în cadrul rețelei de socializare. Cheia primara este id_postare.

RECENZIE = evaluările și părerile utilizatorilor. Cheia primara este id_recenzie.

NOTIFICARE = notificările sau mesajele trimise utilizatorilor în cadrul rețelei de socializare pentru a-i informa despre evenimente, actualizări sau interacțiuni relevante. Cheia primara compusa este alcatuita din id_recenzie, id_postare si id_profil.

CONT = conține informații referitoare la conturile utilizatorilor din rețeaua de socializare, cum ar fi parola. Cheia primara compusa este alcatuita din id_retea, id_utilizator.

RELATII – EX. 4

FONDATOR_creeaza_RETEA_SOCIALA = relatie de tip one-to-many care leaga entitatile FONDATOR si RETEA_SOCIALA, reflectand legatura dintre acestea. Cardinalitatea minima este 1:0 si cardinalitatea maxima este 1:n.

RETEA_SOCIALA_se_afla_la_ADRESA_INFIINTARE = relatie de tip one-to one care leaga entitatile RETEA_SOCIALA si ADRESA_INFIINTARE, reflectand legatura dintre acestea. Cardinalitatea minima/maxima este 1:1.

ADRESA_INFIINTARE_este_LOCALITATE = relatie de tip many-to-one care leaga entitatile ADRESA_INFIINTARE si LOCALITATE, reflectand legatura dintre acestea. Cardinalitatea minima este 1:1 si cardinalitatea maxima este 1:n.

LOCALITATE_se_afla_in_JUDET = relatie de tip many-to-one care leaga entitatile LOCALITATE si JUDET, reflectand legatura dintre acestea. Cardinalitatea minima este 1:1 si cardinalitatea maxima 1:n.

RETEA_SOCIALA_are_DEPARTAMENT = relatie de tip one-to-many care leaga entitatile RETEA_SOCIALA si DEPARTAMENT, reflectand legatura dintre acestea. Cardinalitatea minima este 1:1 si cardinalitatea maxima este 1:n.

DEPARTAMENT_lucreaza_in_ANGAJATI = relatie de tip one-to-many care leaga entitatile DEPARTAMENT si ANGAJATI, reflectand legatura dintre acestea. Cardinalitatea minima este 1:0 si cardinalitatea maxima este 1:n.

ANGAJATI_are_JOB = relatie de tip many-to-one care leaga entitatile ANGAJATI si JOB, reflectand legatura dintre acestea. Cardinalitatea minima este 1:1 si cardinalitatea maxima este 1:n.

RETEA_SOCIALA_are_UTILIZATOR = relatie de tip many-to-many care leaga entitatile RETEA_SOCIALA si UTILIZATOR, reflectand legatura dintre acestea. Cardinalitatea minima este 0:1 si cardinalitatea maxima este n:m.

UTILIZATOR_primeste_PROFIL = relatie de tip many-to-many care leaga entitatile RETEA_SOCIALA si UTILIZATOR, reflectand legatura dintre acestea. Cardinalitatea minima/maxima este 1:1.

PROFIL_are_POSTARE_are_RECENZIE = relatie de tip 3 care leaga entitatile PROFIL, POSTARE si RECENZIE, reflectand legatura dintre acestea. Denumirea acestei relatii o sa fie NOTIFICARE.

ATTRIBUTE – EX. 5

- Entitatea FONDATOR are ca attribute:

id_fondator = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unui fondator.

nume = variabilă de tip caracter, de lungime maximă 100, care reprezintă numele fondatorului.

prenume = variabilă de tip caracter, de lungime maximă 100, care reprezintă prenumele fondatorului.

telefon = variabilă de tip caracter, de lungime maximă 20, care reprezintă numărul de telefon.

- Entitatea JUDET are ca attribute:

id_judet = variabilă de tip întreg, de lungime maximă 20, care reprezintă codul unui judet.

denumire = variabilă de tip caracter, de lungime maximă 100, care reprezintă numele judetului.

- Entitatea LOCALITATE are ca attribute:

id_localitate = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unei localitati.

denumire = variabilă de tip caracter, de lungime maximă 100, care reprezinta numele localitatii.

id_judet = variabilă de tip întreg, de lungime maximă 20, care reprezintă codul judetului din care face parte localitatea. Atributul trebuie sa corespunda la o valoare a cheii primare din tabelul JUDET.

- Entitatea ADRESA_INFIINTARE are ca attribute:

id_adresa = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unei adrese.

strada = variabilă de tip caracter, de lungime maximă 100, care reprezinta strada.

id_localitate = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul localitatii din care face parte adresa. Atributul trebuie sa corespunda la o valoare a cheii primare din tabelul LOCALITATE.

- Entitatea RETEA_SOCIALA are ca attribute:

id_retea = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unei retele.

id_fondator = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unui fondator care a realizat retea. Atributul trebuie sa corespunda la o valoare a cheii primare din tabelul FONDATOR.

denumire = variabilă de tip caracter, de lungime maximă 100, care reprezinta numele retelei.

id_adresa = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul adresei la care se afla retea. Atributul trebuie sa corespunda la o valoare a cheii primare din tabelul ADRESA_INFIINTARE.

- Entitatea JOB are ca attribute:

id_job = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unei job.

salariu = variabila de tip intreg, reprezentand salariul

- Entitatea ANGAJATI are ca attribute:

id_angajat = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unui angajat.

nume = variabilă de tip caracter, de lungime maximă 100, care reprezinta numele angajatului.

prenume = variabilă de tip caracter, de lungime maximă 100, care reprezinta prenumele angajatului.

id_job = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul jobului pe care il are angajatul. Atributul trebuie sa corespunda la o valoare a cheii primare din tabelul JOB.

- Entitatea DEPARTAMENT are ca attribute:

id_departament = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unui department.

denumire = variabilă de tip caracter, de lungime maximă 100, care reprezinta numele departamentului.

id_retea = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul rețelei la care e departamentul. Atributul trebuie sa corespunda la o valoare a cheii primare din tabelul RETEA_SOCIALA.

id_angajat = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul angajatului care lucreaza in departament. Atributul trebuie sa corespunda la o valoare a cheii primare din tabelul ANGAJATI.

- Entitatea UTILIZATORI are ca attribute:

id_utilizator = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unui utilizator.

nume = variabilă de tip caracter, de lungime maximă 100, care reprezinta numele utilizatorului.

prenume = variabilă de tip caracter, de lungime maximă 100, care reprezinta prenumele utilizatorului.

data_nasterii = variabilă de tip dată calendaristică, care reprezintă data nașterii utilizatorului respective.

email = variabilă de tip caracter, de lungime maximă 50, care reprezintă adresa de mail a unui utilizator.

- Entitatea CONT are ca attribute:

id_retea = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul rețelei la care este facut contul. Atributul trebuie sa corespunda la o valoare a cheii primare din tabelul RETEA_SOCIALA.

id_utilizator = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul utilizatorului care detine contul. Atributul trebuie sa corespunda la o valoare a cheii primare din tabelul UTILIZATORI.

parola = variabilă de tip caracter, de lungime maximă 50, care reprezintă parola contului.

- Entitatea PROFIL are ca attribute:

id_profil = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unui profil.

culoare_poza = variabilă de tip caracter, de lungime maximă 30, care reprezintă culoarea pozei de profil.

id_utilizator = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul utilizatorului care detine profilul. Atributul trebuie sa corespunda la o valoare a cheii primare din tabelul UTILIZATORI.

- Entitatea POSTARE are ca attribute:

id_postare = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unei postari.

data_postare = variabilă de tip dată calendaristică, care reprezintă data postarii.

- Entitatea RECENZIE are ca attribute:

id_recenzie = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unei recenzii.

nota = variabilă de tip caracter, de lungime maximă 20, care reprezintă nota.

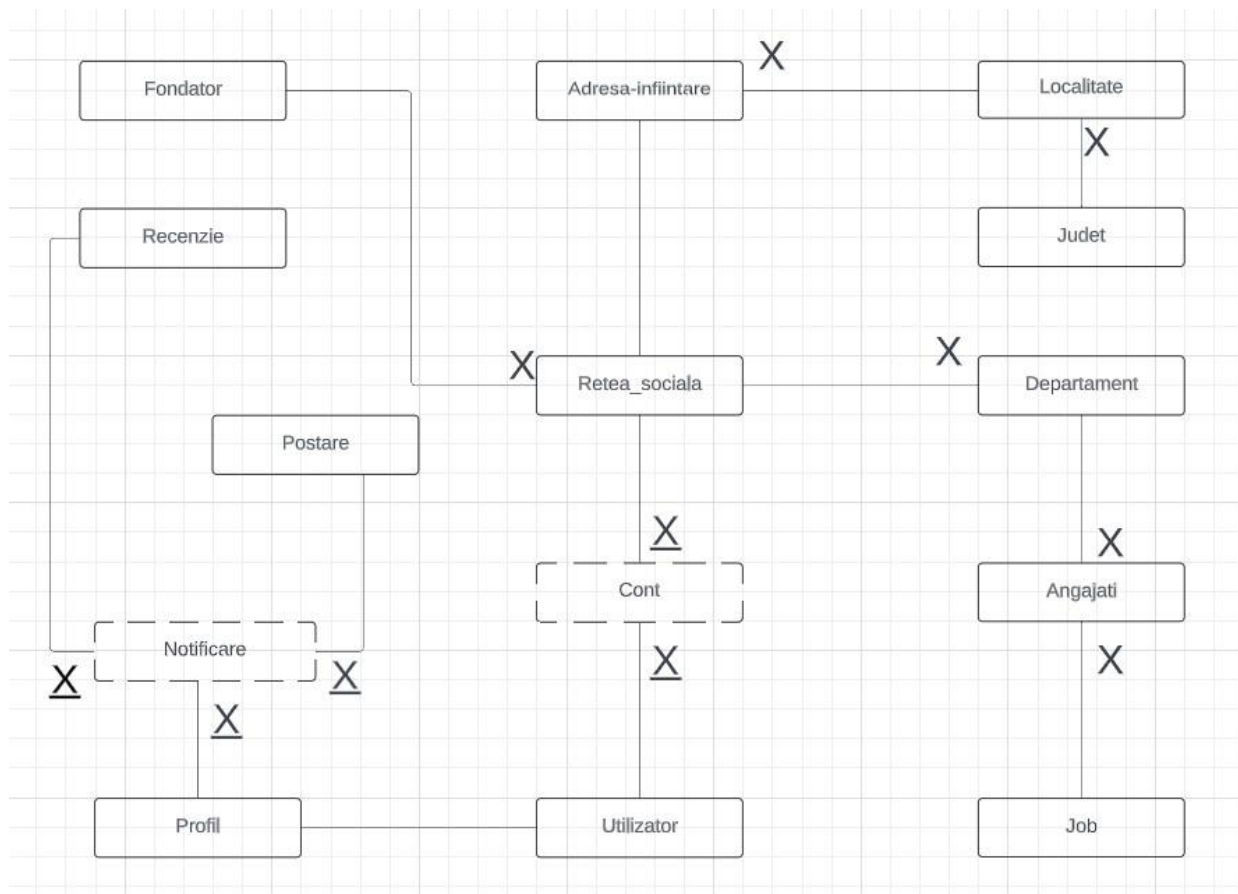
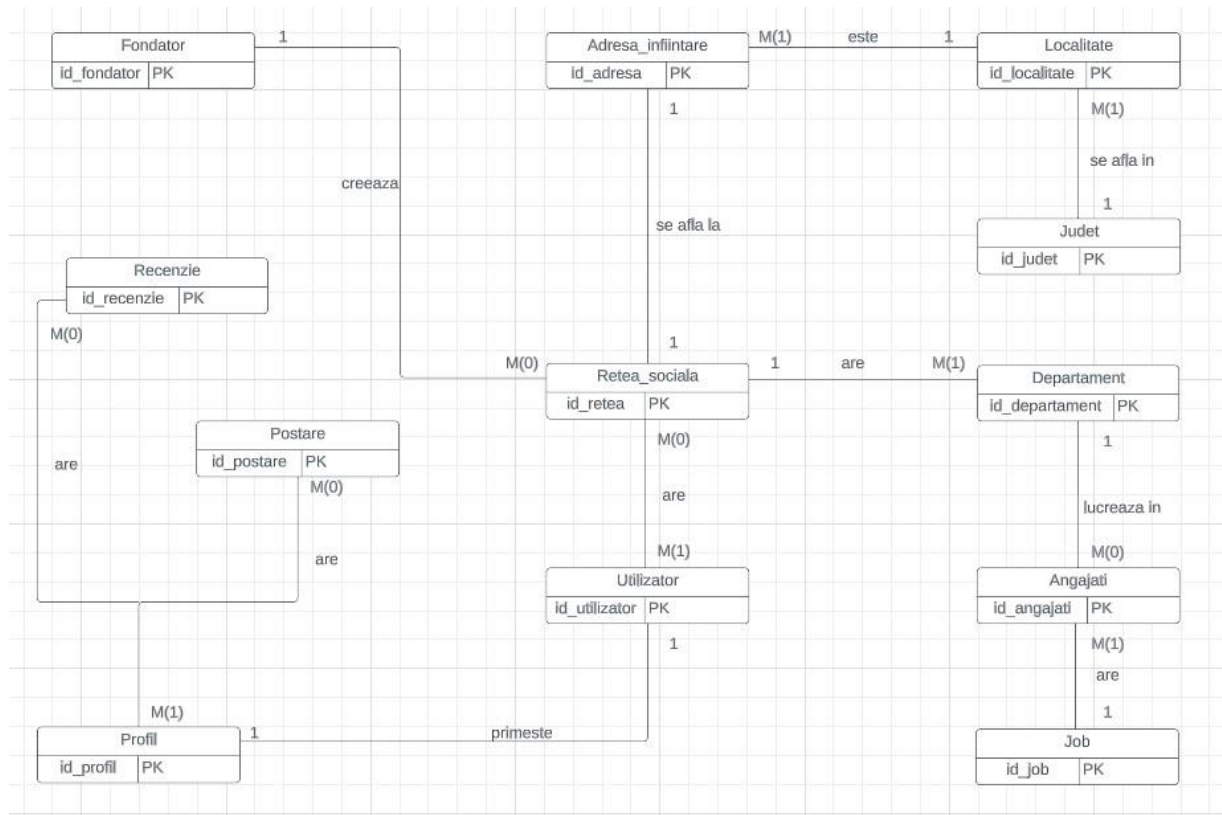
- Entitatea NOTIFICARE are ca attribute:

id_recenzie = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul recenziei. Atributul trebuie sa corespunda la o valoare a cheii primare din tabelul RECENZIE.

id_postare = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul postarii. Atributul trebuie sa corespunda la o valoare a cheii primare din tabelul POSTARE.

id_profil = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul profilului. Atributul trebuie sa corespunda la o valoare a cheii primare din tabelul PROFIL.

ora = variabilă de tip dată calendaristică, care reprezintă data notificarii.



SCHEMELE RELATIONALE – EX. 8

Schemele relationale corespunzatoare diagramei conceptuale sunt:

FONDATOR(id_fondator#, nume, prenume, telefon)

JUDET(id_județ#, denumire)

LOCALITATE(id_localitate#, denumire, id_judet#)

ADRESA_INFIINTARE(id_adresa#, strada, id_localitate#)

RETEA_SOCIALA(id_retea#, id_fondator#, denumire, id_adresa#)

JOB(id_job#, salariu)

ANGAJATI(id_angajat#, nume, prenume, id_job#)

DEPARTAMENT(id_departament#, denumire, id_retea#, id_angajat#)

UTILIZATORI(id_utilizator#, nume prenume, data_nașterii, email)

CONT(id_retea#, id_utilizator#, parolă)

PROFIL(id_profil#, culoare_poza, id_utilizator#)

POSTARE(id_postare#, data_postare)

RECENZIE(id_recenzie#, nota)

NOTIFICARE(id_recenzie#, id_postare#, id_profil#, ora)

NORMALIZARE SI DENORMALIZARE

- Forma normal 1 (FN1)

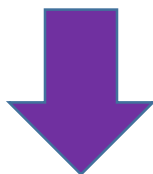
O relatie se afla in FN1 daca fiecarui atribut care are o compunere ii corespunde o valoare indivizibila.

Forma normala 1 este si cea care impune faptul ca fiecare inregistrare sa fie definita incat sa fie identificata unic prin intermediul unei chei primare.

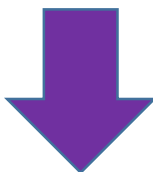
Luam entitatile RETEA si DEPARTAMENT. In cadrul unui retele, denumirea departamentelor nu se repeta, dar in cadrul unui lant de retele acest lucru nu este garantat.

1. Mai multe valori semnifică același camp

RETELE	DEPARTAMENTE
Instagram	Dezvoltare Software, Design
Facebook	Dezvoltare Software, Marketing



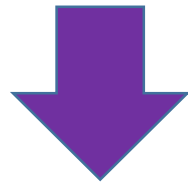
RETELE	DEPARTAMENTE
Instagram	Dezvoltare Software
Instagram	Design
Facebook	Dezvoltare Software
Facebook	Marketing



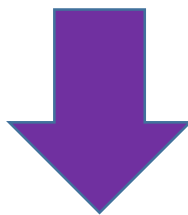
ID	RETELE	DEPARTAMENTE
1	Instagram	Dezvoltare Software
2	Instagram	Design
3	Facebook	Dezvoltare Software
4	Facebook	Marketing

2. Mai multe coloane reprezintă același tip de date.

RETELE	DEPARTAMENTE	DEPARTAMENTE(1)
Instagram	Dezvoltare Software	Design
Facebook	Dezvoltare Software	Marketing



RETELE	DEPARTAMENTE
Instagram	Dezvoltare Software
Facebook	Dezvoltare Software
Instagram	Design
Facebook	Marketing



ID	RETELE	DEPARTAMENTE
1	Instagram	Dezvoltare Software
2	Instagram	Design
3	Facebook	Dezvoltare Software
4	Facebook	Marketing

Pentru a asigura unicitatea unei inregistrari, se va utiliza cheia primara. In exemplul de mai sus, prin introducerea unei coloane aditionale de tip intreg se asigura unicitatea fiecarei inregistrari.

- Forma Normală 2 (FN2)

O relație se află în a doua formă normală dacă și numai dacă această relație este deja în FN1 și fiecare atribut care nu este cheie primară este dependent de întreaga cheie primară.

FN2 interzice existența dependențelor funcționale parțiale în cadrul relației.

Dacă unul sau mai multe elemente sunt dependente funcționale numai de o parte a cheii primare, atunci ele trebuie să fie separate în tabele diferite. Dacă tabela are o cheie primară formată din numai un atribut, atunci ea este automat în FN2.

Exemplificare pentru diagrama CONT:

id_retea#	id_utilizator#	nume utilizator	parola
R1	U1	N1	adknhac
R2	U2	N2	pogrbrg
53	U3	N3	jtdntnd

O rețea poate să aibă mai mulți utilizatori, iar un utilizator poate să fie pe mai multe rețele.

Relația este în FN1. (avem identificator unic pentru toate intrările din tabel)

Fiecare atribut care nu este cheie (nu participă la cheia primară) este dependent de întreaga cheie primară, în cazul nostru atributul nume utilizator, nu este cheie și trebuie să depindă direct de întreaga cheie primară $id_rete\#$ și $id_utilizator\#$ -> aceste atribute nu depind direct de întreaga cheie primară deoarece se observă dependența directă dintre nume utilizator și $id_utilizator$, însemnând că nume utilizator depinde direct doar de o parte a cheii primare, și anume doar de $id_utilizator$ -> relația nu se află în FN2.

Astfel avem ca:

- $\{id_utilizator\# \} \rightarrow \{nume_utilizator\}$ – $id_utilizator$ determină funcțional $nume_utilizator$
- $\{id_utilizator\#, id_rete\#\} \rightarrow \{parola\}$

Pentru a avea relația în FN2, numele doctorului trebuie să fie doar în entitatea doctor (se aplică regula Casey Delobel).

$id_rete\#$	$id_utilizator\#$	nume utilizator	parola
R1	U1	N1	adknhac
R2	U2	N2	pogrbrg
R3	U3	N3	jtdntnd



$id_rete\#$	$id_utilizator$	parola
R1	U1	adknhac
R2	U2	pogrbrg
R3	U3	jtdntnd

- Forma Normala 3(FN3)

O relatie este in a treia forma normala daca si numai daca este in FN2 si fiecare atribut care nu este cheie depinde direct de cheia primara.

Initial tabelul DEPARTAMENT ar avea urmatoarea forma:

id_departament#	denumire_departament	nume_angajat	prenume_angajat
D1	DE1	N1	P1
D2	DE2	N2	P2
D3	DE3	N3	P3

Se poate observa ca atribut prenume_angajat depinde de atributul nume_angajat care depinde la randul sau de cheia primara id_departament, altfel nefiind in FN3.

Pentru a aduce in FN3, separ attributele din tabelul DEPARTAMENT, aparand astfel tabela ANGAJATI.

Astfel, inlocuiesc aceste attribute cu cheia straina id_departament (aparand astfel o relatie one-to-many intre DEPARTAMENT si ANGAJATI) pentru a determina mai usor din ce department face parte angajatul.

id_departament	denumire	id_angajat	id_retea
D1	DE1	A1	R1
D2	DE2	A2	R2
D3	DE3	A3	R3

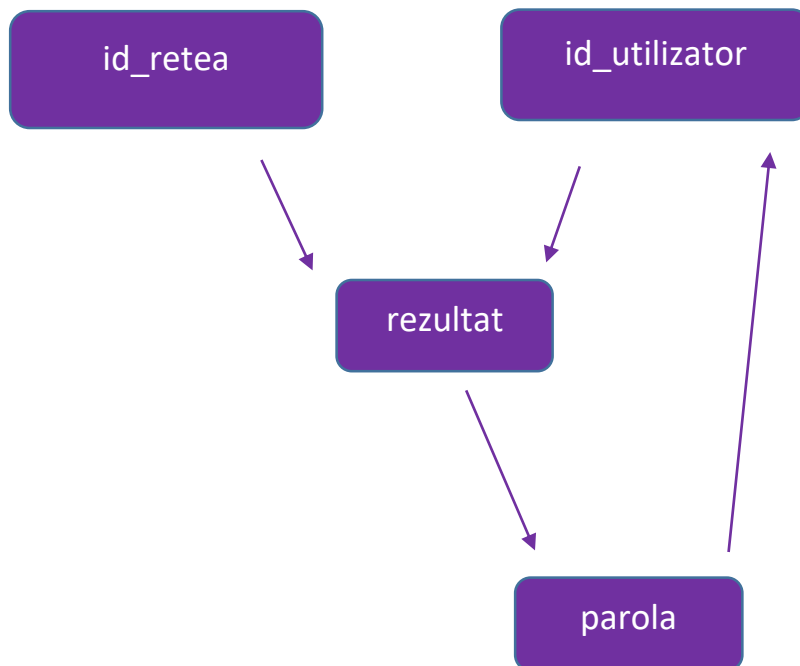
id_angajat	nume	prenume	id_job	id_departament
A1	N1	P1	J1	D1
A2	N2	P2	J2	D2
A3	N3	P3	J3	D3

- Forma normala Boyce-Codd (BCNF)

Forma normala Boyce-Codd se bazeaza pe dependentele functionale care iau in considerare toate cheile candidat dintr-o relatie.

Pentru relatiile cu o singura cheie candidat, formele FN3 si BCNF sunt echivalente (FONDATOR, JOB etc.)

Vom lua relatia relatie CONT(id_retea#, id_utilizator#, parola)



Regula Casey Delobel pentru CONT(id_retea#, id_utilizator#, parola) ne arata ca rezultat determina functional pe id_utilizator.

Rezultat->id_utilizator.

- CONT_1(id_retea, rezultat)
- CONT_2(rezultat, id_utilizator)

- Forma normală 4 (FN4)

FN4 elimina redundantele datorate relatiilor m:n, adica datorate dependentei multiple.

O relatie este in a patra forma normala daca si numai daca este in BCNF si nu contine relatii m:n independente.

Se ia entitatea FONDATOR(id_fondator#, nume, prenume, telefon) si presupun ca un fondator poate avea mai multe nume, prenume si mai multe numere de telefon

id_fondator# -> nume

id_fondator# -> prenume

id_fondator# -> telefon

Relatia FONDATOR este in BCNF. Pentru a aduce relatia in FN4 o vom descompune pe proiectie in trei relatii: FONDATOR1(id_fondator#, nume) FONDATOR2(id_fondator#, prenume) FONDATOR3(id_fondator#, telefon)

->FONDATOR= JOIN(FONDATOR1.FONDATOR2,FONDATOR3)

- Forma normala 5(FN5)

O relatie R este in FN5 (numita si forma normala proiectie-uniune) daca si numai daca orice dependenta de uniune a lui R este o consecinta a unui chei candidat a lui R.

Orice relatie care este in FN5 este si in FN4, deoarece fiecare dependenta multivaloare poate fi privita ca un caz particular de dependenta de uniune.

Orice relatie poate fi descompusa fara pierderi la uniune intr-o multime de relatii care sunt in FN5. Pentru a preciza daca o relatie este in FN5, este suficient sa cunoastem cheile candidate si toate dependentele de uniune din R.

Aducerea in FN5 presupune eliminarea join dependentelor.

Voi lua relatia UTILIZATORI si presupun ca am o join dependenta in aceasta. Voi lua multimea (nume, prenume, email) si presupun ca exista multiple dependente intre fiecare dintre perechile din multimi (nume, email), (nume, prenume), (prenume, email).

- Denormalizarea

Denormalizarea este procesul invers al procesului de normalizare. Denormalizarea functioneaza adaugand date redundante sau grupand date pentru a optimiza performanta.

Denormalizare are rolul de a realiza executarea mai rapidă a interogărilor prin introducerea redundanței, punând accentul pe rapiditatea analizei și scăzând numărul de tabele.

Motivul efectuării denormalizării este dat de costul produs în procesor de interogare printr-o structură supra-normalizată.

Denormalizarea poate fi definită și ca metoda de stocare a îmbinării relațiilor de formă normală superioară ca relație de bază, care se află într-o formă normală inferioară. Reduce numărul de tabele și îmbinările de tabel complicate, deoarece un număr mai mare de îmbinări poate încetini procesul.

Aici abordarea denormalizării, subliniază conceptul că, plasând toate datele într-un singur loc, ar putea elimina necesitatea căutării acelor fișiere multiple pentru a colecta aceste date.

În cadrul bazei mele de date, luând spre exemplu, relația NOTIFICARE și descompunerea ei de la FN4, este inutil și mult mai costisitor din punct de vedere al timpului de executare să parcurgem datele și din NOTIFICARE _1 și din NOTIFICARE _2.

TABELE

```
CREATE TABLE FONDATOR(  
    id_fondator NUMBER(5) CONSTRAINT PKEY_FONDATOR PRIMARY KEY,  
    nume VARCHAR(100) CONSTRAINT nume_fondator NOT NULL,  
    prenume VARCHAR(100) CONSTRAINT prenume_fondator NOT NULL,  
    telefon VARCHAR(20) CONSTRAINT telefon_student NOT NULL  
);
```

```
INSERT INTO FONDATOR  
VALUES(1, 'Popa', 'Ana', '0766658432');
```

```
INSERT INTO FONDATOR  
VALUES(2, 'Popescu', 'Dan', '0765423453');
```

```
INSERT INTO FONDATOR  
VALUES(3, 'Rizea', 'Andrada', '0745437687');
```

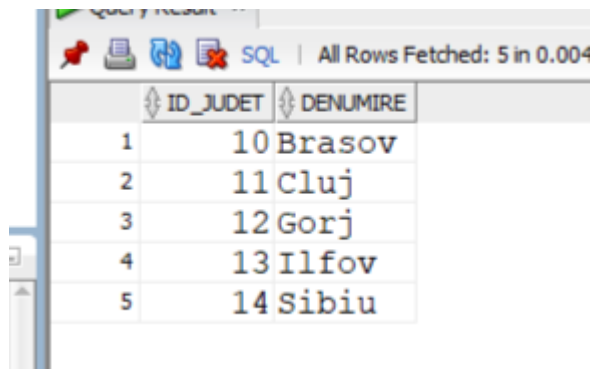
```
INSERT INTO FONDATOR  
VALUES(4, 'Ionescu', 'Andrei', '0765473896');
```

```
INSERT INTO FONDATOR  
VALUES(5, 'Lazar', 'Matei', '0783742854');
```



	ID_FONDATOR	NUME	PRENUME	TELEFON
1	1	Popa	Ana	0766658432
2	2	Popescu	Dan	0765423453
3	3	Rizea	Andrada	0745437687
4	4	Ionescu	Andrei	0765473896
5	5	Lazar	Matei	0783742854

```
CREATE TABLE JUDET(  
    id_judet NUMBER(30) CONSTRAINT PKEY_JUDET PRIMARY KEY,  
    denumire VARCHAR(100) CONSTRAINT denumire_judet NOT NULL  
);  
  
INSERT INTO JUDET  
VALUES(10, 'Brasov');  
  
INSERT INTO JUDET  
VALUES(11, 'Cluj');  
  
INSERT INTO JUDET  
VALUES(12, 'Gorj');  
  
INSERT INTO JUDET  
VALUES(13, 'Ilfov');  
  
INSERT INTO JUDET  
VALUES(14, 'Sibiu');
```



The screenshot shows a SQL query result window with the title 'Query Results'. The status bar at the top indicates 'All Rows Fetched: 5 in 0.004'. The table has two columns: 'ID_JUDET' and 'DENUMIRE'. The data is as follows:

ID_JUDET	DENUMIRE
1	10 Brasov
2	11 Cluj
3	12 Gorj
4	13 Ilfov
5	14 Sibiu


```
CREATE TABLE LOCALITATE(  
    id_localitate NUMBER(5) CONSTRAINT PKEY_LOCALITATE PRIMARY KEY,  
    denumire VARCHAR(100) CONSTRAINT denumire_localitate NOT NULL,  
    id_judet NUMBER(30),  
    CONSTRAINT fk_localitate FOREIGN KEY(id_judet) REFERENCES JUDET(id_judet)  
);
```

```
INSERT INTO LOCALITATE
```

```
VALUES(20, 'Brasov', 10);
```

```
INSERT INTO LOCALITATE
```

```
VALUES(21, 'Cluj-Napoca', 11);
```

```
INSERT INTO LOCALITATE
```

```
VALUES(22, 'Dej', 11);
```

```
INSERT INTO LOCALITATE
```

```
VALUES(23, 'Tg-Jiu', 12);
```

```
INSERT INTO LOCALITATE
```

```
VALUES(24, 'Buftea', 13);
```

```
INSERT INTO LOCALITATE
```

```
VALUES(25, 'Sibiu', 14);
```



The screenshot shows a database query result with 6 rows. The columns are ID_LOCALITATE, DENUMIRE, and ID_JUDET. The data is as follows:

	ID_LOCALITATE	DENUMIRE	ID_JUDET
1	20	Brasov	10
2	21	Cluj-Napoca	11
3	22	Dej	11
4	23	Tg-Jiu	12
5	24	Buftea	13
6	25	Sibiu	14

Baze de date

Vatau Lorena

Grupa 133

```
CREATE TABLE ADRESA_INFIINTARE(
```

```
    id_adresa NUMBER(5) CONSTRAINT PKEY_ADRESA_INFIINTARE PRIMARY KEY,
```

```
    strada VARCHAR(100) CONSTRAINT adresa_strada NOT NULL,
```

```
    id_localitate NUMBER(5),
```

```
    CONSTRAINT fk_adresa_infiintare FOREIGN KEY(id_localitate) REFERENCES LOCALITATE(id_localitate)
```

```
);
```

```
INSERT INTO ADRESA_INFIINTARE
```

```
VALUES(30, 'Calea Poienii', 20);
```

```
INSERT INTO ADRESA_INFIINTARE
```

```
VALUES(31, 'Aleea Orhideei', 21);
```

```
INSERT INTO ADRESA_INFIINTARE
```

```
VALUES(32, 'Strada Amurg', 22);
```

```
INSERT INTO ADRESA_INFIINTARE
```

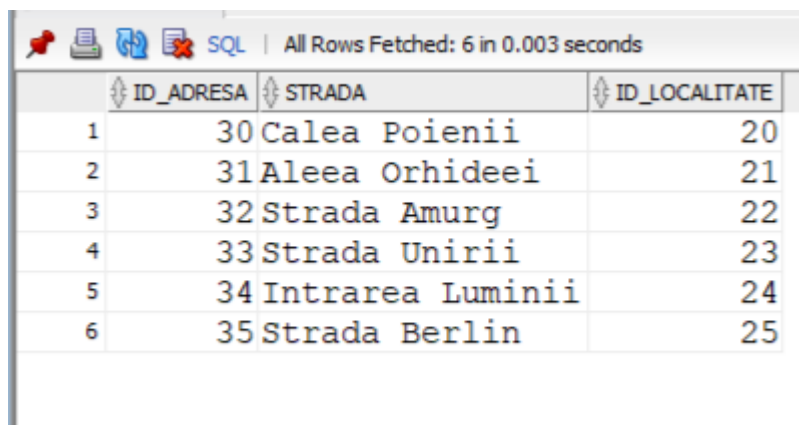
```
VALUES(33, 'Strada Unirii', 23);
```

```
INSERT INTO ADRESA_INFIINTARE
```

```
VALUES(34, 'Intrarea Luminii', 24);
```

```
INSERT INTO ADRESA_INFIINTARE
```

```
VALUES(35, 'Strada Berlin', 25);
```



The screenshot shows a database query result with 6 rows. The columns are ID_ADRESA, STRADA, and ID_LOCALITATE. The data is as follows:

	ID_ADRESA	STRADA	ID_LOCALITATE
1	30	Calea Poienii	20
2	31	Aleea Orhideei	21
3	32	Strada Amurg	22
4	33	Strada Unirii	23
5	34	Intrarea Luminii	24
6	35	Strada Berlin	25

Baze de date

Vatau Lorena

Grupa 133

CREATE TABLE RETEA_SOCIALA (

id_retea NUMBER(5) CONSTRAINT PKEY_RETEA_SOCIALA PRIMARY KEY,

id_fondator NUMBER(5),

denumire VARCHAR(100) NOT NULL,

id_adresa NUMBER(5)

);

ALTER TABLE RETEA_SOCIALA

ADD CONSTRAINT fk_id_fondator FOREIGN KEY(id_fondator) REFERENCES FONDATOR(id_fondator);

ALTER TABLE RETEA_SOCIALA

ADD CONSTRAINT fk_id_adresa FOREIGN KEY(id_adresa) REFERENCES ADRESA_INFIINTARE(id_adresa);

INSERT INTO RETEA_SOCIALA

VALUES(100, 1, 'Instagram', 30);

INSERT INTO RETEA_SOCIALA

VALUES(101, 2, 'Facebook', 31);

INSERT INTO RETEA_SOCIALA

VALUES(102, 3, 'Snapchat', 32);

INSERT INTO RETEA_SOCIALA

VALUES(103, 4, 'Tiktok', 33);

INSERT INTO RETEA_SOCIALA

VALUES(104, 5, 'Whatsapp', 34);

SQL All Rows Fetched: 5 in 0.004 seconds				
	ID_RETEA	ID_FONDATOR	DENUMIRE	ID_ADRESA
1	100	1	Instagram	30
2	101	2	Facebook	31
3	102	3	Snapchat	32
4	103	4	Tiktok	33
5	104	5	Whatsapp	34

```
CREATE TABLE JOB(
    id_job NUMBER(5) CONSTRAINT PKEY_id_job PRIMARY KEY,
    salariu NUMBER(5) CONSTRAINT salariu NOT NULL
);
```

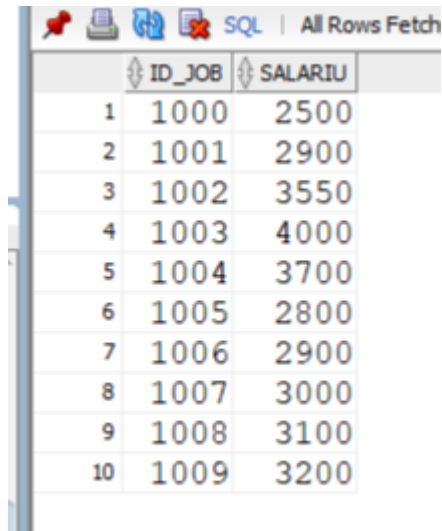
```
INSERT INTO JOB
VALUES(1000, 2500);
```

```
INSERT INTO JOB
VALUES(1001, 2900);
```

```
INSERT INTO JOB
VALUES(1002, 3550);
```

```
INSERT INTO JOB
VALUES(1003, 4000);
```

```
INSERT INTO JOB
VALUES(1004, 3700);
```



	ID_JOB	SALARIU
1	1000	2500
2	1001	2900
3	1002	3550
4	1003	4000
5	1004	3700
6	1005	2800
7	1006	2900
8	1007	3000
9	1008	3100
10	1009	3200

```
CREATE TABLE ANGAJATI(  
    id_angajat NUMBER(5) CONSTRAINT PKEY_id_angajat PRIMARY KEY,  
    nume VARCHAR(100) CONSTRAINT nume_angajat NOT NULL,  
    prenume VARCHAR(100) CONSTRAINT prenume_angajat NOT NULL,  
    id_job NUMBER(5),  
    CONSTRAINT fk_angajati FOREIGN KEY(id_job) REFERENCES JOB(id_job)  
);
```

```
ALTER TABLE angajati
```

```
ADD id_departament NUMBER(5);
```

```
ALTER TABLE ANGAJATI
```

```
ADD CONSTRAINT fk_ang FOREIGN KEY(id_departament) REFERENCES DEPARTAMENT(id_departament);
```

```
SELECT*
```

```
FROM ANGAJATI;
```

```
INSERT INTO ANGAJATI
```

```
VALUES( 50, 'Florea', 'Denisa', 1000);
```

```
INSERT INTO ANGAJATI  
VALUES( 51, 'Igneu', 'Traian', 1001);
```

```
INSERT INTO ANGAJATI  
VALUES( 52, 'Munteanu', 'Daria', 1002);
```

```
INSERT INTO ANGAJATI  
VALUES( 53, 'Olteanu', 'Claudiu', 1003);
```

```
INSERT INTO ANGAJATI  
VALUES( 54, 'Radu', 'Silvia', 1004);
```

```
UPDATE ANGAJATI  
SET id_departament = 60  
WHERE id_angajat=50;
```

```
UPDATE ANGAJATI  
SET id_departament = 61  
WHERE id_angajat=51;
```

```
UPDATE ANGAJATI  
SET id_departament = 62  
WHERE id_angajat=52;
```

```
UPDATE ANGAJATI  
SET id_departament = 63  
WHERE id_angajat=53;
```

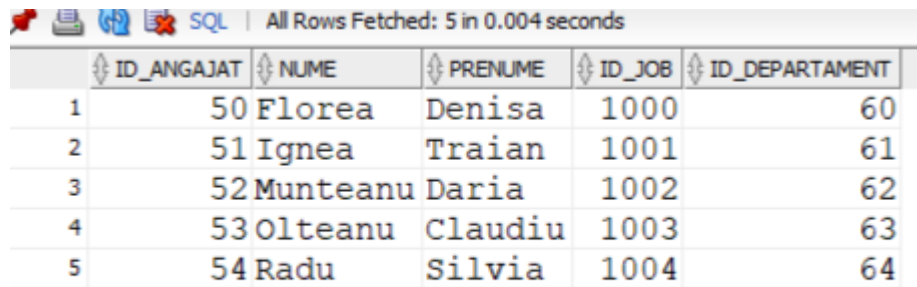
```
UPDATE ANGAJATI  
SET id_departament = 64
```

Baze de date

Vatau Lorena

Grupa 133

WHERE id_angajat=54;



ID_ANGAJAT	NUME	PRENUME	ID_JOB	ID_DEPARTAMENT
1	50 Florea	Denisa	1000	60
2	51 Ignea	Traian	1001	61
3	52 Munteanu	Daria	1002	62
4	53 Olteanu	Claudiu	1003	63
5	54 Radu	Silvia	1004	64

```
CREATE TABLE DEPARTAMENT(
```

```
    id_departament NUMBER(5) CONSTRAINT PKEY_DEPARTAMENT PRIMARY KEY,
```

```
    denumire VARCHAR(100) CONSTRAINT denumire_departament NOT NULL,
```

```
    id_retea NUMBER(5),
```

```
    id_angajat NUMBER(5)
```

```
);
```

```
ALTER TABLE DEPARTAMENT
```

```
ADD CONSTRAINT fk_id_retea FOREIGN KEY(id_retea) REFERENCES RETEA_SOCIALA(id_retea);
```

```
ALTER TABLE DEPARTAMENT
```

```
ADD CONSTRAINT fk_id_angajat FOREIGN KEY(id_angajat) REFERENCES ANGAJATI(id_angajat);
```

```
INSERT INTO DEPARTAMENT
```

```
VALUES(60, 'Dezvoltare Produs', 100, 50);
```

```
INSERT INTO DEPARTAMENT
```

```
VALUES(61, 'Marketing', 101, 51);
```

```
INSERT INTO DEPARTAMENT
```


```
VALUES(62, 'Suport Tehnic', 102, 52);
```

```
INSERT INTO DEPARTAMENT
```

```
VALUES(63, 'Securitate', 103, 53);
```

```
INSERT INTO DEPARTAMENT
```

```
VALUES(64, 'Comunitate si Moderare', 104, 54);
```

 | All Rows Fetched: 5 in 0.017 seconds

	ID_DEPARTAMENT	DENUMIRE	ID_RETEA	ID_ANGAJAT
1	60	Dezvoltare Produs	100	50
2	61	Marketing	101	51
3	62	Suport Tehnic	102	52
4	63	Securitate	103	53
5	64	Comunitate si Moderare	104	54

```
CREATE TABLE UTILIZATORI(
```

```
    id_utilizator NUMBER(5) CONSTRAINT PKEY_UTILIZATORI PRIMARY KEY,
```

```
    nume VARCHAR(100) CONSTRAINT nume_utilizator NOT NULL,
```

```
    prenume VARCHAR(100) CONSTRAINT prenume_utilizator NOT NULL,
```

```
    data_nasterii DATE CONSTRAINT data_nasterii NOT NULL,
```

```
    email VARCHAR(50) UNIQUE
```

```
);
```

```
INSERT INTO UTILIZATORI
```

```
VALUES(200, 'Boltasu', 'Bogdan', TO_DATE('12.03.1992', 'dd.mm.yyyy'), 'bboltasu@gmail.com');
```

```
INSERT INTO UTILIZATORI
```

```
VALUES(201, 'Cernat', 'Cezar', TO_DATE('10.03.1987', 'dd.mm.yyyy'), 'cezarc@gmail.com');
```

```
INSERT INTO UTILIZATORI
```


Baze de date

Vatau Lorena

Grupa 133

```
VALUES(202, 'Nicoara', 'Ada', TO_DATE('08.09.1999', 'dd.mm.yyyy'), 'adaaa@gmail.com');
```

```
INSERT INTO UTILIZATORI
```

```
VALUES(203, 'Muresan', 'Bianca', TO_DATE('20.10.1972', 'dd.mm.yyyy'), 'bbianca@gmail.com');
```

```
INSERT INTO UTILIZATORI
```

```
VALUES(204, 'Parfene', 'Sorin', TO_DATE('16.12.1985', 'dd.mm.yyyy'), 'sorin@gmail.com');
```

```
INSERT INTO UTILIZATORI
```

```
VALUES(205, 'Popescu', 'Valentin', TO_DATE('29.11.1997', 'dd.mm.yyyy'), 'valentin@gmail.com');
```

```
INSERT INTO UTILIZATORI
```

```
VALUES(206, 'Radu', 'Ramona', TO_DATE('22.08.1990', 'dd.mm.yyyy'), 'ramona@gmail.com');
```

```
INSERT INTO UTILIZATORI
```

```
VALUES(207, 'Silaghi', 'Andrei', TO_DATE('17.06.1993', 'dd.mm.yyyy'), 'silaghia@gmail.com');
```

```
INSERT INTO UTILIZATORI
```

```
VALUES(208, 'Stanecu', 'Oana', TO_DATE('23.11.1989', 'dd.mm.yyyy'), 'oana@gmail.com');
```

```
INSERT INTO UTILIZATORI
```

```
VALUES(209, 'Stoian', 'Loredana', TO_DATE('15.05.1999', 'dd.mm.yyyy'), 'loredana@gmail.com');
```

```
INSERT INTO UTILIZATORI
```

```
VALUES(210, 'Tuca', 'Vanesa', TO_DATE('19.08.1996', 'dd.mm.yyyy'), 'vanesa@gmail.com');
```

SQL | All Rows Fetched: 11 in 0.014 seconds

	ID_UTILIZATOR	NUME	PRENUME	DATA_NASTERII	EMAIL
1	200	Florea	Bogdan	12-MAR-92	bboltasu@gmail.com
2	201	Cernat	Cezar	10-MAR-87	cezarc@gmail.com
3	202	Nicoara	Ada	08-SEP-99	adaaaa@gmail.com
4	203	Muresan	Bianca	20-OCT-72	bbianca@gmail.com
5	204	Parfene	Sorin	16-DEC-85	sorin@gmail.com
6	205	Popescu	Valentin	29-NOV-97	valentin@gmail.com
7	206	Radu	Ramona	22-AUG-90	ramona@gmail.com
8	207	Silaghi	Andrei	17-JUN-93	silaghia@gmail.com
9	208	Stanecu	Oana	23-NOV-89	oana@gmail.com
10	209	Stoian	Loredana	15-MAY-99	loredana@gmail.com
11	210	Tuca	Vanesa	19-AUG-96	vanesa@gmail.com

```
CREATE TABLE CONT(
    id_retea NUMBER(5) CONSTRAINT pk_id_retea REFERENCES RETEA_SOCIALA(id_retea),
    id_utilizator NUMBER(5) CONSTRAINT pk_id_utilizator REFERENCES UTILIZATORI(id_utilizator),
    parola VARCHAR(50) CONSTRAINT parola NOT NULL,
    CONSTRAINT pk_compus_cont primary key(id_retea,id_utilizator)
);
```

```
INSERT INTO CONT
VALUES(100, 200, 'aaabbbc');
```

```
INSERT INTO CONT
VALUES(100, 201, 'daaadsbbc');
```

```
INSERT INTO CONT
VALUES(101, 202, 'gagebbbc');
```

```
INSERT INTO CONT
VALUES(101, 203, 'dbuwqk');
```

Baze de date

Vatau Lorena

Grupa 133

```
INSERT INTO CONT
```

```
VALUES(102, 204, 'sfrvv');
```

```
INSERT INTO CONT
```

```
VALUES(102, 205, 'SJBDWXNIJ');
```

```
INSERT INTO CONT
```

```
VALUES(103, 206, 'tyndujt');
```

```
INSERT INTO CONT
```

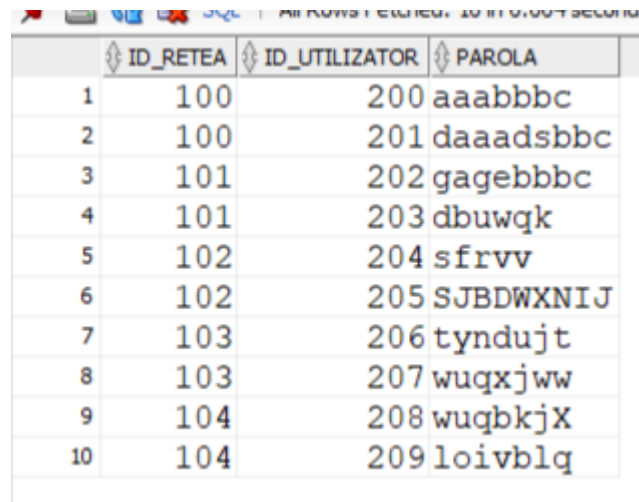
```
VALUES(103, 207, 'wuqxjww');
```

```
INSERT INTO CONT
```

```
VALUES(104, 208, 'wuqbkjX');
```

```
INSERT INTO CONT
```

```
VALUES(104, 209, 'loivblq');
```



	ID_RETEA	ID_UTILIZATOR	PAROLA
1	100	200	aaabbbc
2	100	201	daaadsbbc
3	101	202	gagebbbc
4	101	203	dbuwqk
5	102	204	sfrvv
6	102	205	SJBDWXNIJ
7	103	206	tyndujt
8	103	207	wuqxjww
9	104	208	wuqbkjX
10	104	209	loivblq

```
CREATE TABLE PROFIL(  
    id_profil NUMBER(5) CONSTRAINT PKEY_PROFIL PRIMARY KEY,  
    culoare_poza VARCHAR(30) CONSTRAINT culoare_poza NOT NULL,  
    id_utilizator NUMBER(5),  
    CONSTRAINT fk_profil FOREIGN KEY(id_utilizator) REFERENCES UTILIZATORI(id_utilizator)  
);
```

```
INSERT INTO PROFIL  
VALUES(300, 'albastru', 200);
```

```
INSERT INTO PROFIL  
VALUES(301, 'alb', 201);
```

```
INSERT INTO PROFIL  
VALUES(302, 'galben', 202);
```

```
INSERT INTO PROFIL  
VALUES(303, 'roz', 203);
```

```
INSERT INTO PROFIL  
VALUES(304, 'rosu', 204);
```

```
INSERT INTO PROFIL  
VALUES(305, 'mov', 205);
```

```
INSERT INTO PROFIL  
VALUES(306, 'verde', 206);
```

```
INSERT INTO PROFIL
```

Baze de date

Vatau Lorena

Grupa 133

```
VALUES(307, 'portocaliu', 207);
```

```
INSERT INTO PROFIL
```

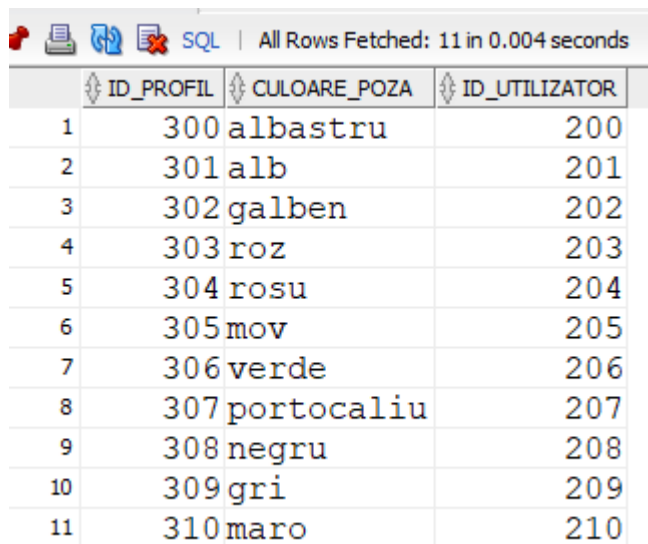
```
VALUES(308, 'negru', 208);
```

```
INSERT INTO PROFIL
```

```
VALUES(309, 'gri', 209);
```

```
INSERT INTO PROFIL
```

```
VALUES(310, 'maro', 210);
```



The screenshot shows a database query result with 11 rows. The columns are ID_PROFIL, CULOARE_POZA, and ID_UTILIZATOR. The data is as follows:

	ID_PROFIL	CULOARE_POZA	ID_UTILIZATOR
1	300	albastru	200
2	301	alb	201
3	302	galben	202
4	303	roz	203
5	304	rosu	204
6	305	mov	205
7	306	verde	206
8	307	portocaliu	207
9	308	negru	208
10	309	gri	209
11	310	maro	210

```
CREATE TABLE POSTARE(
```

```
    id_postare NUMBER(5) CONSTRAINT PKEY_POSTARE PRIMARY KEY,
```

```
    data_postare DATE CONSTRAINT data_postare NOT NULL
```

```
);
```

```
INSERT INTO POSTARE
```

```
VALUES(900, TO_DATE('29.11.2022', 'dd.mm.yyyy'));
```

Baze de date

Vatau Lorena

Grupa 133

```
INSERT INTO POSTARE
```

```
VALUES(901, TO_DATE('20.11.2022', 'dd.mm.yyyy'));
```

```
INSERT INTO POSTARE
```

```
VALUES(902, TO_DATE('10.09.2022', 'dd.mm.yyyy'));
```

```
INSERT INTO POSTARE
```

```
VALUES(903, TO_DATE('07.01.2023', 'dd.mm.yyyy'));
```

```
INSERT INTO POSTARE
```

```
VALUES(904, TO_DATE('20.03.2023', 'dd.mm.yyyy'));
```

```
INSERT INTO POSTARE
```

```
VALUES(905, TO_DATE('23.11.2022', 'dd.mm.yyyy'));
```

```
INSERT INTO POSTARE
```

```
VALUES(906, TO_DATE('17.05.2022', 'dd.mm.yyyy'));
```

```
INSERT INTO POSTARE
```

```
VALUES(907, TO_DATE('01.04.2023', 'dd.mm.yyyy'));
```

```
INSERT INTO POSTARE
```

```
VALUES(908, TO_DATE('10.10.2022', 'dd.mm.yyyy'));
```

```
INSERT INTO POSTARE
```

```
VALUES(909, TO_DATE('19.02.2023', 'dd.mm.yyyy'));
```

	ID_POSTARE	DATA_POSTARE
1	900	29-NOV-22
2	901	20-NOV-22
3	902	10-SEP-22
4	903	07-JAN-23
5	904	20-MAR-23
6	905	23-NOV-22
7	906	17-MAY-22
8	907	01-APR-23
9	908	10-OCT-22
10	909	19-FEB-23

```
CREATE TABLE RECENZIE(  
    id_recenzie NUMBER(5) CONSTRAINT PKEY_RECENZIE PRIMARY KEY,  
    nota VARCHAR(20) CONSTRAINT nota NOT NULL  
);
```

```
INSERT INTO RECENZIE  
VALUES(800, '10');
```

```
INSERT INTO RECENZIE  
VALUES(801, '7');
```

```
INSERT INTO RECENZIE  
VALUES(802, '9');
```

```
INSERT INTO RECENZIE  
VALUES(803, '10');
```

```
INSERT INTO RECENZIE  
VALUES(804, '5');
```

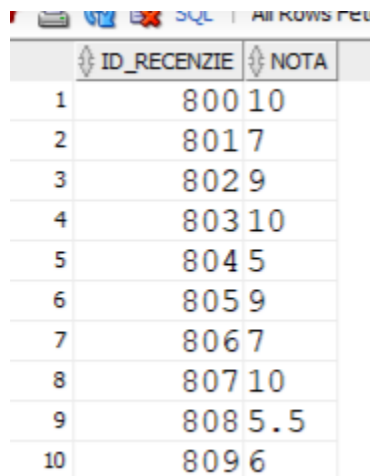
```
INSERT INTO RECENZIE  
VALUES(805, '9');
```

```
INSERT INTO RECENZIE  
VALUES(806, '7');
```

```
INSERT INTO RECENZIE  
VALUES(807, '10');
```

```
INSERT INTO RECENZIE  
VALUES(808, '5');
```

```
INSERT INTO RECENZIE  
VALUES(809, '6');
```



The screenshot shows a database window with a table named RECENZIE. The table has two columns: ID_RECENZIE and NOTA. The data is as follows:

	ID_RECENZIE	NOTA
1	800	10
2	801	7
3	802	9
4	803	10
5	804	5
6	805	9
7	806	7
8	807	10
9	808	5.5
10	809	6

```
CREATE TABLE NOTIFICARE(  
    id_recenzie NUMBER(5) CONSTRAINT pk_RECENZIE REFERENCES RECENZIE(id_recenzie),  
    id_postare NUMBER(5) CONSTRAINT pk_POSTARE REFERENCES POSTARE(id_postare),  
    id_profil NUMBER(5) CONSTRAINT pk_PROFIL REFERENCES PROFIL(id_profil),
```



```
ora DATE CONSTRAINT ora NOT NULL,  
  
CONSTRAINT pk_compus_not primary key(id_recenzie,id_postare, id_profil)  
  
);
```

```
INSERT INTO NOTIFICARE  
VALUES(800,900,300, TO_DATE('22.05.2023 15:30:00', 'dd.mm.yyyy HH24:MI:SS'));
```

```
INSERT INTO NOTIFICARE  
VALUES(801,901,301, TO_DATE('22.04.2023 13:20:00', 'dd.mm.yyyy HH24:MI:SS'));
```

```
INSERT INTO NOTIFICARE  
VALUES(802,902,302, TO_DATE('12.09.2022 16:38:00', 'dd.mm.yyyy HH24:MI:SS'));
```

```
INSERT INTO NOTIFICARE  
VALUES(803,903,303, TO_DATE('26.06.2022 10:30:00', 'dd.mm.yyyy HH24:MI:SS'));
```

```
INSERT INTO NOTIFICARE  
VALUES(804,904,304, TO_DATE('22.05.2023 19:30:00', 'dd.mm.yyyy HH24:MI:SS'));
```

```
INSERT INTO NOTIFICARE  
VALUES(805,905,305, TO_DATE('29.10.2022 14:00:00', 'dd.mm.yyyy HH24:MI:SS'));
```

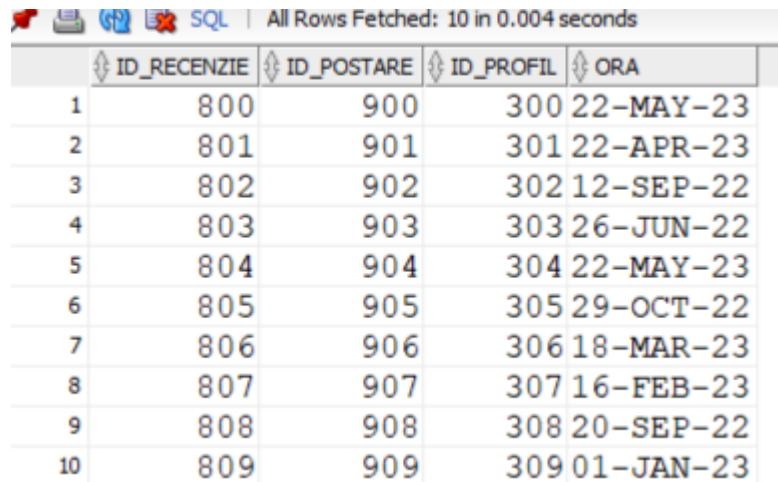
```
INSERT INTO NOTIFICARE  
VALUES(806,906,306, TO_DATE('18.03.2023 17:45:50', 'dd.mm.yyyy HH24:MI:SS'));
```

```
INSERT INTO NOTIFICARE  
VALUES(807,907,307, TO_DATE('16.02.2023 12:20:10', 'dd.mm.yyyy HH24:MI:SS'));
```

```
INSERT INTO NOTIFICARE  
VALUES(808,908,308, TO_DATE('20.09.2022 20:35:40', 'dd.mm.yyyy HH24:MI:SS'));
```

```
INSERT INTO NOTIFICARE
```

```
VALUES(809,909,309, TO_DATE('01.01.2023 16:39:11', 'dd.mm.yyyy HH24:MI:SS'));
```



SQL | All Rows Fetched: 10 in 0.004 seconds

	ID_RECENZIE	ID_POSTARE	ID_PROFIL	ORA
1	800	900	300	22-MAY-23
2	801	901	301	22-APR-23
3	802	902	302	12-SEP-22
4	803	903	303	26-JUN-22
5	804	904	304	22-MAY-23
6	805	905	305	29-OCT-22
7	806	906	306	18-MAR-23
8	807	907	307	16-FEB-23
9	808	908	308	20-SEP-22
10	809	909	309	01-JAN-23

Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele – ex 10

```
CREATE SEQUENCE salary_increment
```

```
INCREMENT BY 100
```

```
START WITH 2500
```

```
MAXVALUE 4500
```

```
CYCLE;
```

```
INSERT INTO JOB
```

```
VALUES (1005,salary_increment.nextval);
```

```
INSERT INTO JOB
```

```
VALUES (1006, salary_increment.nextval);
```

```
INSERT INTO JOB
```

```
VALUES (1007, salary_increment.nextval);
```

```
INSERT INTO JOB
```

```
VALUES (1008, salary_increment.nextval);
```

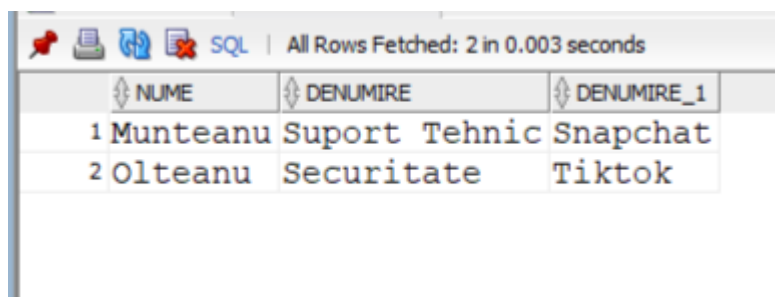
```
INSERT INTO JOB
```

```
VALUES (1009, salary_increment.nextval);
```

CERERI SQL – EX. 12

--1. afisati numele angajatilor, numele departamentului, numele retelei unde numele angajatului contine
--sirul "te"

```
SELECT a.numa, d.denumire, r.denumire  
FROM angajati a, departament d, retea_sociala r  
WHERE a.id_departament = d.id_departament  
AND d.id_retea = r.id_retea  
AND lower(a.numa) LIKE '%te%';
```

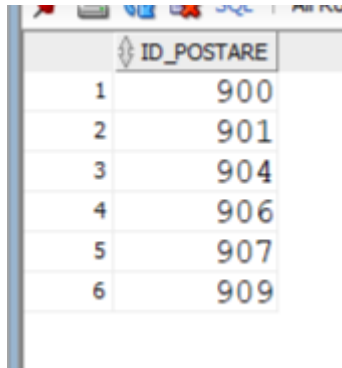


	NUME	DENUMIRE	DENUMIRE_1
1	Munteanu Suport Tehnic		Snapchat
2	Olteanu Securitate		Tiktok

--2. afisati codul postarii pentru toate înregistrările care au o înregistrare corespunzătoare
--în tabela notificare din anul 2023

```
SELECT po.id_postare
```

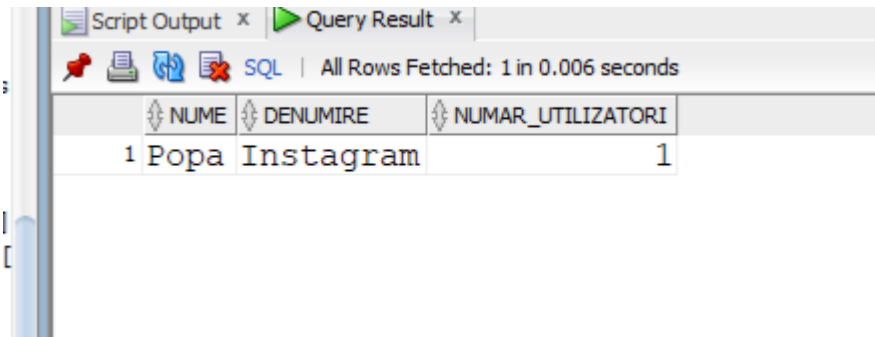
```
FROM postare po, (SELECT id_postare
FROM notificare
WHERE to_char(ora, 'yyyy') = '2023') n
WHERE po.id_postare = n.id_postare;
```



	ID_POSTARE
1	900
2	901
3	904
4	906
5	907
6	909

--3. Afisati numele fondatorului, denumirea retelei sociale si numarul de utilizatori înregistrati în rețeaua sociala
--pentru fondatorii cu sediul în orasul Brasov, afisând doar acele
--grupuri care au cel puțin un utilizator înregistrat

```
SELECT f.num, r.denumire, NVL(COUNT(*), 0) AS numar_utilizatori
FROM fondator f, retea_sociala r, adresa_infiintare a, localitate l, judet jdt
WHERE f.id_fondator = r.id_fondator
AND r.id_adresa = a.id_adresa
AND a.id_localitate = l.id_localitate
AND l.id_judet = jdt.id_judet
AND initcap(jdt.denumire) = 'Brasov'
GROUP BY f.num, r.denumire
HAVING COUNT(*) >= 1;
```

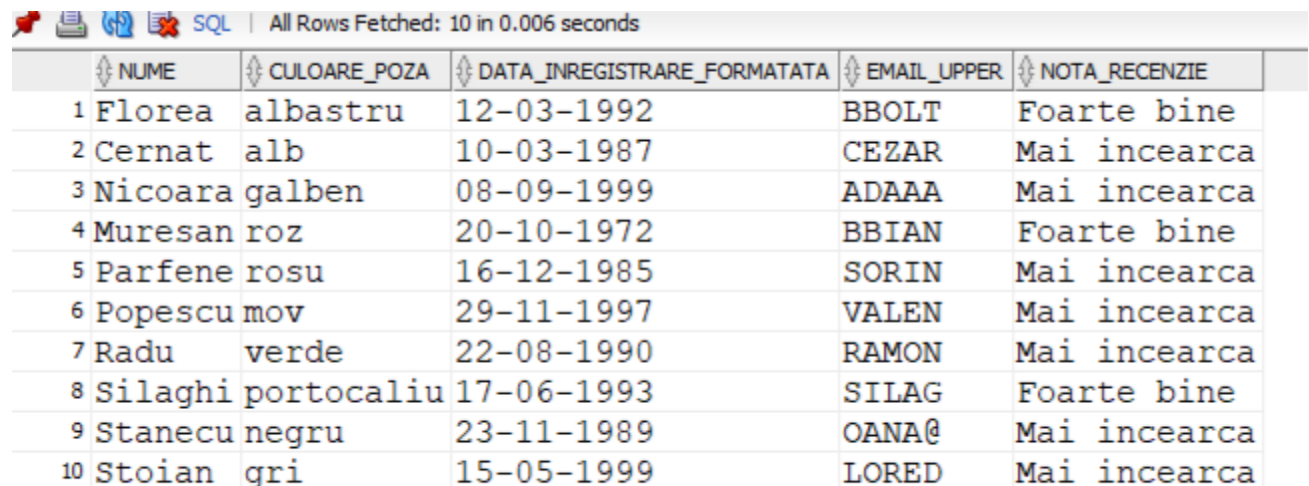


	NUME	DENUMIRE	NUMAR_UTILIZATORI
1	Popa	Instagram	1

--4. Obtine numele utilizatorilor, culoarea pozei de profil, data de înregistrare

--formatati, primele 5 caractere ale adresei de email convertite în majuscule si o descriere a recenziei bazata

--pe nota acordata. Rezultatele trebuie sortate în functie de nume.



	NUME	CULOARE_POZA	DATA_INREGISTRARE_FORMATATA	EMAIL_UPPER	NOTA_RECENZIE
1	Florea	albastru	12-03-1992	BBOLT	Foarte bine
2	Cernat	alb	10-03-1987	CEZAR	Mai incearca
3	Nicoara	galben	08-09-1999	ADAAA	Mai incearca
4	Muresan	roz	20-10-1972	BBIAN	Foarte bine
5	Parfene	rosu	16-12-1985	SORIN	Mai incearca
6	Popescu	mov	29-11-1997	VALEN	Mai incearca
7	Radu	verde	22-08-1990	RAMON	Mai incearca
8	Silaghi	portocaliu	17-06-1993	SILAG	Foarte bine
9	Stanecu	negru	23-11-1989	OANA@	Mai incearca
10	Stoian	gri	15-05-1999	LORED	Mai incearca

```
SELECT u.num, p.culoare_poza, TO_CHAR(u.data_nasterii, 'DD-MM-YYYY') AS data_inregistrare_formatata,
UPPER(SUBSTR(u.email, 1, 5)) AS email_upper,
CASE
    WHEN to_char(r.nota) = '10' THEN 'Foarte bine'
    WHEN to_char(r.nota)='6' THEN 'Mai incearca'
    ELSE 'Mai incearca'
END AS nota_recenzie
```

```
FROM utilizatori u, profil p, notificare n, recenzie r, cont c

WHERE

    u.id_utilizator = p.id_utilizator

    AND u.id_utilizator = c.id_utilizator

    AND n.id_recenzie=r.id_recenzie

    AND n.id_profil=p.id_profil;

ORDER BY u.num;
```

--5. Sa se afiseze numarul total de retele sociale înregistrate în fiecare localitate si

--retea sociala , adresa si localitate. Pentru fiecare localitate, se va afisa denumirea localitatii

--si numarul de retele sociale înregistrate în acea localitate

```
WITH adrese_retea AS (

    SELECT r.id_retea, a.id_adresa, l.denumire AS nume_localitate

    FROM retea_sociala r, adresa_infiintare a, localitate l

    WHERE r.id_adresa = a.id_adresa

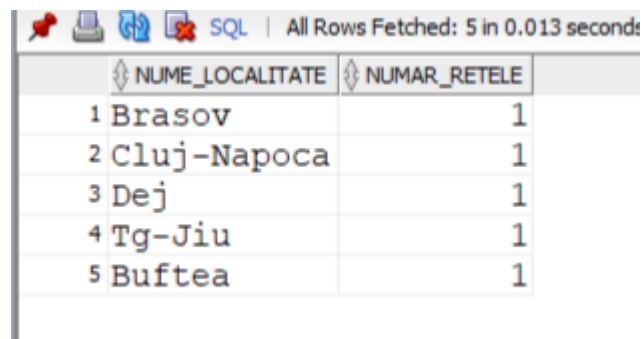
    AND a.id_localitate = l.id_localitate

)

SELECT ar.nume_localitate, COUNT(*) AS numar_retele

FROM adrese_retea ar

GROUP BY ar.nume_localitate;
```



	NUME_LOCALITATE	NUMAR_RETELE
1	Brasov	1
2	Cluj-Napoca	1
3	Dej	1
4	Tg-Jiu	1
5	Buftea	1

UPDATE-URI SI SUPRIMARI– EX. 13

```
UPDATE utilizatori  
SET nume = (SELECT nume  
            FROM angajati  
            WHERE id_angajat=50)  
WHERE id_utilizator = 200;
```

```
SELECT*  
FROM UTILIZATORI;
```

```
UPDATE RECENZIE  
SET nota = nota + 0.5  
WHERE nota < 7 and id_recenzie in (select id_recenzie  
                                   from notificare  
                                   where to_char(ora, 'yyyy') = '2022');
```

```
ALTER TABLE JOB  
ADD salariu_min NUMBER(5);
```

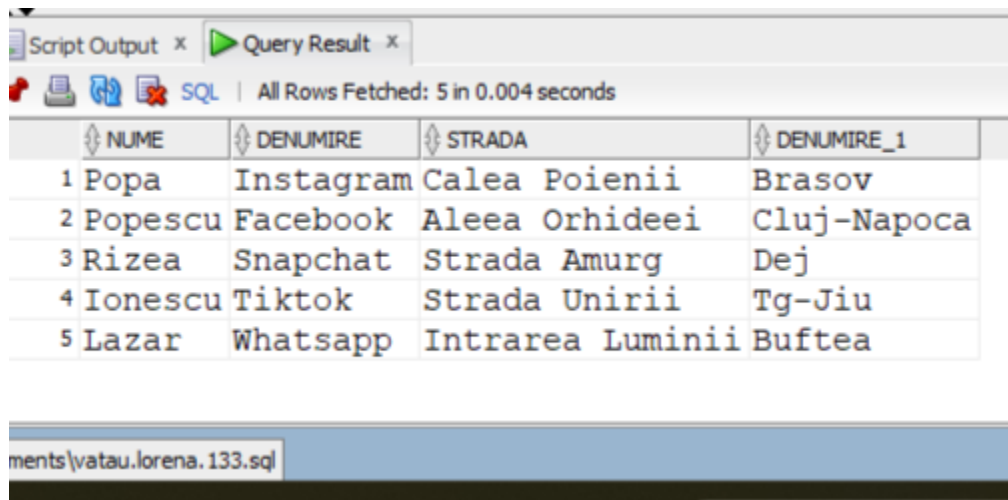
```
SELECT*  
FROM JOB;
```

```
ALTER TABLE JOB  
DROP COLUMN salariu_min;
```

EX. 15

- a) Obțineti numele fondatorului, denumirea rețelei sociale, strada adresei
- ai denumirea localitati. Daca exista o potrivire înregistrata între tabele, aflați valorile corespunzatoare.
- Daca nu exista o potrivire înregistrata, afiați valori nule pentru
- câmpurile din tabelul corespunzător utilizând outer join

```
SELECT f.num, r.denumire, a.strada, l.denumire
FROM fondator f, retea_sociala r, adresa_infiintare a, localitate l
WHERE f.id_fondator = r.id_fondator (+)
      AND r.id_adresa = a.id_adresa (+)
      AND a.id_localitate = l.id_localitate (+);
```



The screenshot shows a SQL query result window with the following data:

	NUME	DENUMIRE	STRADA	DENUMIRE_1
1	Popa	Instagram	Calea Poienii	Brasov
2	Popescu	Facebook	Aleea Orhideei	Cluj-Napoca
3	Rizea	Snapchat	Strada Amurg	Dej
4	Ionescu	Tiktok	Strada Unirii	Tg-Jiu
5	Lazar	Whatsapp	Intrarea Luminii	Buftea

- b) Afiseaza toate codurile retelelor pentru care fiecare adresa are o localitate valida
- si fiecare localitate are un judea asociat.

```
SELECT r.id_retea
FROM retea_sociala r
```


Baze de date

Vatau Lorena

Grupa 133

WHERE NOT EXISTS (

SELECT *

FROM adresa_infiintare a

WHERE NOT EXISTS (

SELECT *

FROM localitate l

WHERE NOT EXISTS (

SELECT *

FROM judet j

WHERE j.id_judet = l.id_judet

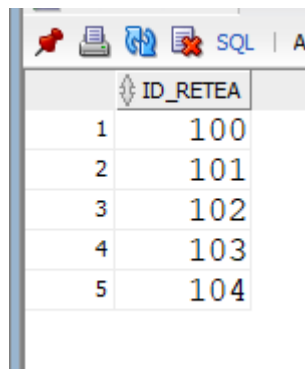
AND l.id_localitate = a.id_localitate

AND a.id_adresa = r.id_adresa

)

)

);



	ID_RETEA
1	100
2	101
3	102
4	103
5	104

--c) Afiseaza departamentele si angajatii din acele departamente care

--au mai putin de 2 angajati în acelasi departament.

SELECT d.denumire AS nume_departament, a.num AS nume_angajat

FROM angajati a, departament d

WHERE a.id_departament = d.id_departament

AND (

Baze de date

Vatau Lorena

Grupa 133

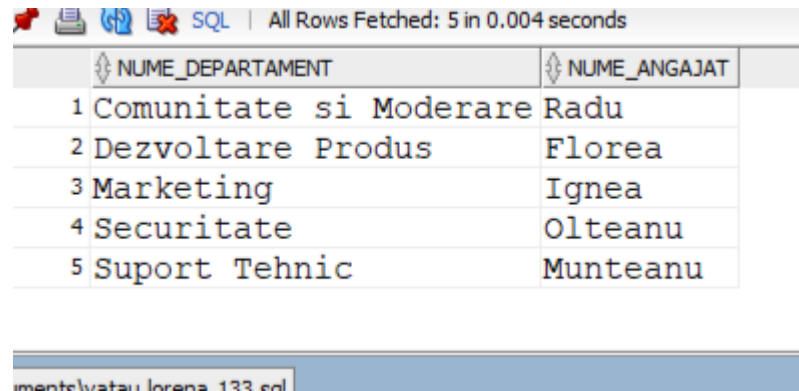
```
SELECT COUNT(*)
```

```
FROM angajati a2
```

```
WHERE a2.id_departament = a.id_departament
```

```
) < 2
```

```
ORDER BY d.denumire;
```



	NUME_DEPARTAMENT	NUME_ANGAJAT
1	Comunitate si Moderare	Radu
2	Dezvoltare Produs	Florea
3	Marketing	Ignea
4	Securitate	Olteanu
5	Suport Tehnic	Munteanu

mentelvatau lorena 133 col