

# PROIECT BAZE DE DATE

## 1.Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.

O companie își propune să administreze mai multe rețele sociale. Fiecare rețea socială va beneficia, în urma parteneriatului cu această companie, de un administrator, de suport tehnic oferit de programatori și de moderarea interacțiunii dintre utilizatori.

Fiecare utilizator își poate deschide conturi pe mai multe rețele sociale, unde poate scrie postări și comenta postările altor utilizatori. Sarcina unui moderator este de a se asigura că postările și comentariile respectă regulile de etichetă ale rețelei sociale. Pentru a simplifica sarcina moderatorului, se va reține locația acestuia și, când este posibil, a utilizatorilor, astfel încât moderatorul să poată interacționa cu utilizatori din aceeași zonă geografică.

De asemenea, utilizatorii pot oferi feedback unei rețele sociale, pentru a ajuta la îmbunătățirea serviciilor oferite.

## 2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.

Informațiile de contact ale persoanelor trebuie să fie unice.

Un programator poate avea următoarele tipuri de job: frontend, backend.

Un utilizator poate fi asociat cu mai mulți utilizatori.

Pentru postări și comentarii, se va reține data la care au fost scrise.

## 3. Descrierea entităților, incluzând precizarea cheii primare.

ENTITATE	CHEIE PRIMARĂ	OBSERVAȚII
Administrator	ID_Admin	Persoana ce ia deciziile principale cu privire la administrarea și dezvoltarea unei rețele sociale
Rețea	ID_Network	Date despre fiecare rețea socială aflată în colaborare cu firma
Feedback	ID_Feedback	Părerea oferită opțional de fiecare utilizator
Utilizator	ID_User	Date despre utilizatorii rețelor sociale
Programator	ID_Programmer	Specialist care asigură suport tehnic
Postare	ID_Post	Date despre postările utilizatorilor
Comentariu	ID_Comment	Date despre comentariile utilizatorilor
Locație	ID_Location	Informații ce determină o zonă geografică
Moderator	ID_Modea	Persoană ce asigură un mediu ideal pentru comunicarea decentă dintre utilizatorii

4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

RELATIE	CARDINALITATE	OBSERVAȚII
Administrator – Rețea	one-to-many	Un admin poate manageria mai multe rețele. O rețea este manageriată de un administrator.
Rețea – Utilizator	many-to-many	O rețea are mai mulți utilizatori. Un utilizator este prezent pe mai multe rețele.
Rețea – Feedback	one-to-many	O rețea poate primi mai multe feedback-uri. Un feedback aparține unei rețele.
Feedback – Utilizator	one-to-many	Un utilizator poate scrie mai multe feedback-uri. Un feedback este scris de un utilizator.
Utilizator – Utilizator	one-to-many	Un utilizator poate fi asociat cu mai mulți utilizatori. Un utilizator este asociat cu un utilizator.
Rețea – Programator	many-to-many	O rețea are mai mulți programatori. Un programator poate lucra la mai multe rețele.
Utilizator – Postare	one-to-many	Un utilizator scrie mai multe postări. O postare este scrisă de un utilizator.
Utilizator – Comentariu	one-to-many	Un utilizator poate scrie un comentariu. Un comentariu este scris de un utilizator.
Utilizator – Locație	one-to-many	O locație poate avea mai mulți utilizatori. Un utilizator poate avea asociată o locație.
Locație – Moderator	one-to-many	O locație poate avea mai mulți moderatori. Un moderator are asociată o locație.
Moderator – Postare	many-to-many	Un moderator poate analiza mai multe postări. O postare poate fi analizată de mai mulți mods.
Moderator – Comentariu	many-to-many	Un moderator poate analiza mai multe comentarii. Un comentariu poate fi analizat de mai mulți mods.

5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.

ENTITATE: ADMINISTRATOR

ATRIBUT	TIP	DEFAULT	NOT NULL	OBSERVAȚII
ID_Admin	Int	Auto_increment	True	-
Nume	Varchar	-	True	-
Prenume	Varchar	-	True	-
Contact	Varchar	-	True	Valoare unică

ENTITATE: REȚEA

ATRIBUT	TIP	DEFAULT	NOT NULL	OBSERVAȚII
ID_Network	Int	Auto_increment	True	-
ID_Admin	Int	-	True	Cheie străină către administrator
Denumire	Varchar	-	True	-
Contact	Varchar	-	True	Valoare unică

ENTITATE: FEEDBACK

ATRIBUT	TIP	DEFAULT	NOT NULL	OBSERVAȚII
ID_Feedback	Int	Auto_increment	True	-
ID_Network	Int	-	True	Cheie străină către rețea
ID_User	Int	-	True	Cheie străină către utilizator
Text	Varchar	-	True	Conținutul feedback-ului

ENTITATE: UTILIZATOR

ATRIBUT	TIP	DEFAULT	NOT NULL	OBSERVAȚII
ID_User	Int	Auto_increment	True	-
ID_Friend	Int	-	False	Relație recursivă
ID_Location	Int	-	False	Cheie străină către locație
Nume	Varchar	-	True	-
Prenume	Varchar	-	True	-

ENTITATE: CONT

ATRIBUT	TIP	DEFAULT	NOT NULL	OBSERVAȚII
ID_Account	Int	Auto_increment	True	-
ID_Network	Int	-	True	Cheie străină către rețea
ID_User	Int	-	False	Cheie străină către utilizator
Username	Varchar	-	True	Nume ales de utilizator pentru cont
Email	Varchar	-	True	Valoare unică

ENTITATE: PROGRAMATOR

ATRIBUT	TIP	DEFAULT	NOT NULL	OBSERVAȚII
ID_Programmer	Int	Auto_increment	True	-
Nume	Varchar	-	True	-
Prenume	Varchar	-	True	-
Titlu_job	Varchar	frontend	True	Valori: frontend, backend

ENTITATE: ACTUALIZARE

ATRIBUT	TIP	DEFAULT	NOT NULL	OBSERVAȚII
ID_Update	Int	Auto_increment	True	-
ID_Network	Int	-	True	Cheie străină către rețea
ID_Programmer	Int	-	True	Cheie străină către programator
Details	Varchar	-	True	Ce aduce nou actualizarea

ENTITATE: COMENTARIU

ATRIBUT	TIP	DEFAULT	NOT NULL	OBSERVAȚII
ID_Comment	Int	Auto_increment	True	-
ID_Account	Int	-	True	Cheie străină către cont
Text	Varchar	-	True	Conținutul text
Data	Date	sysdate	True	Data la care a fost scris

ENTITATE: POSTARE

ATRIBUT	TIP	DEFAULT	NOT NULL	OBSERVAȚII
ID_Post	Int	Auto_increment	True	-
ID_Account	Int	-	True	Cheie străină către cont
Titlu	Varchar	-	False	Titlul postării
Text	Varchar	-	True	Conținutul text
Data	Date	sysdate	True	Data la care a fost scrisă

ENTITATE: LOCAȚIE

ATRIBUT	TIP	DEFAULT	NOT NULL	OBSERVAȚII
ID_Locație	Int	Auto_increment	True	-
Stat	Varchar	-	True	-
Oraș	Varchar	-	False	-

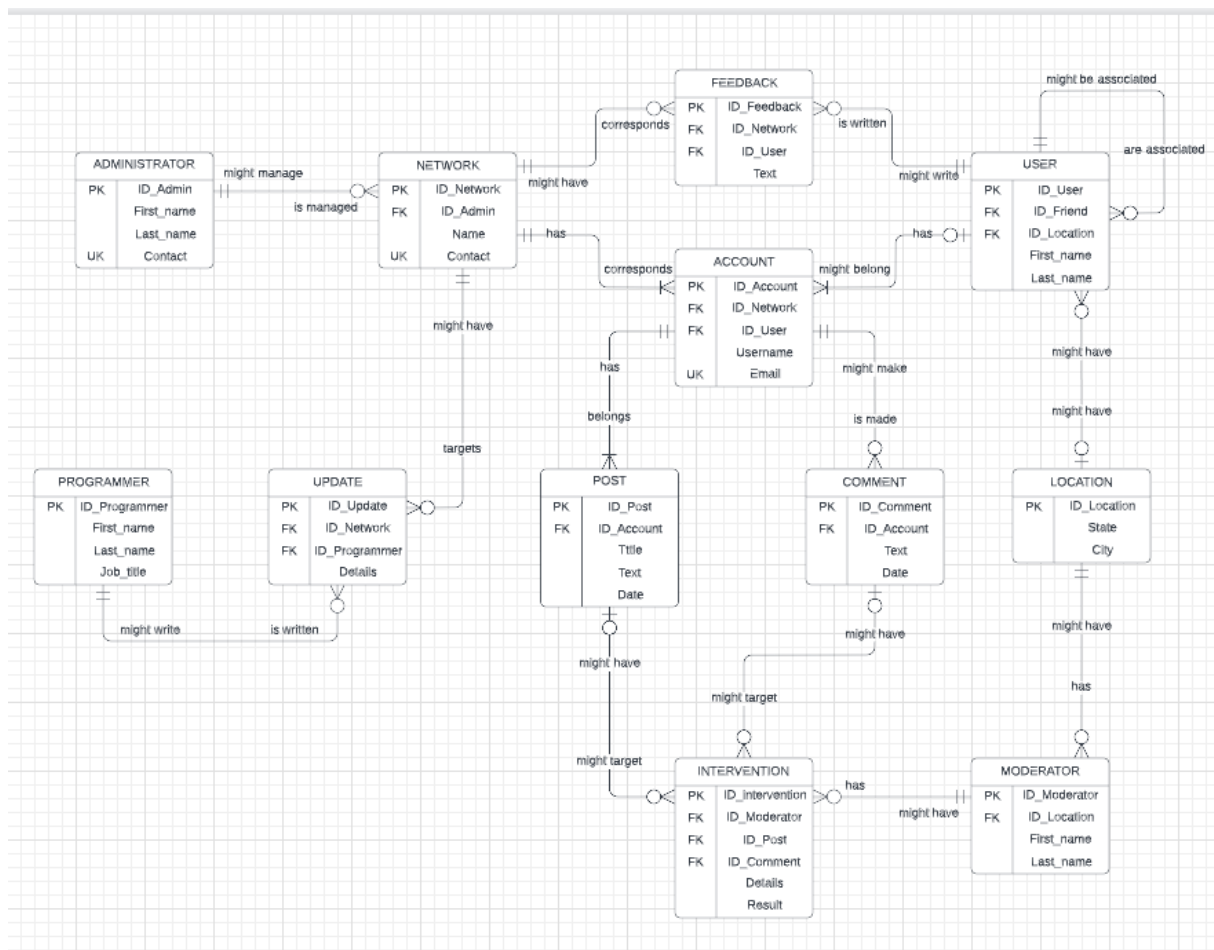
ENTITATE: INTERVENȚIE

ATRIBUT	TIP	DEFAULT	NOT NULL	OBSERVAȚII
ID_Intervention	Int	Auto_increment	True	-
ID_Moderator	Int	-	True	Cheie străină către moderator
ID_Post	Int	-	False	Cheie străină către postare
ID_Comment	Int	-	False	Cheie străină către comentariu
Detalii	Varchar	-	True	Informații despre intervenție
Rezultat	Varchar	-	False	Consecințele intervenției

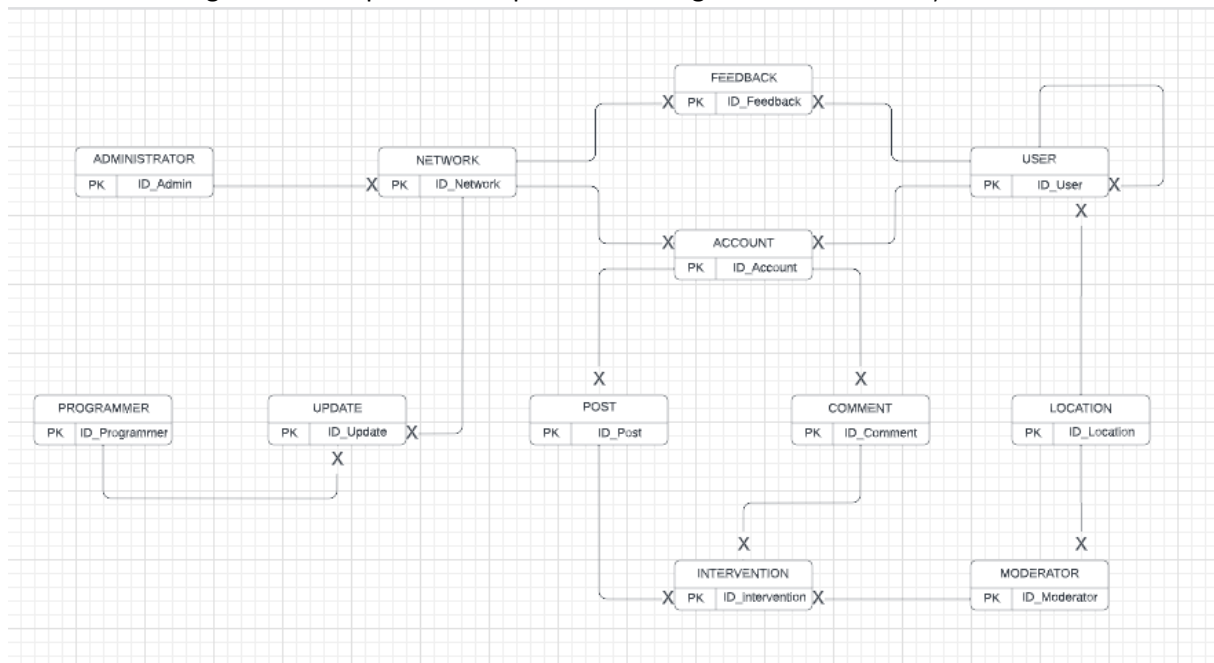
ENTITATE: MODERATOR

ATRIBUT	TIP	DEFAULT	NOT NULL	OBSERVAȚII
ID_Moderator	Int	Auto_increment	True	-
ID_Location	Int	-	True	Cheie străină către locație
Nume	Varchar	-	True	-
Prenume	Varchar	-	True	-

## 6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



## 7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație



8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale de la punctul 7.

ADMINISTRATOR (#ID\_Admin, Nume, Prenume, Contact)

REȚEA (#ID\_Network, ID\_Admin, Denumire, Contact)

FEEDBACK (#ID\_Feedback, ID\_Network, ID\_User, Text)

UTILIZATOR (#ID\_User, ID\_Friend, ID\_Location, Nume, Prenume)

CONT (#ID\_Account, ID\_Network, ID\_User, Username, Email)

PROGRAMATOR (#ID\_Programmer, Nume, Prenume, Titlu\_job)

ACTUALIZARE (#ID\_Update, ID\_Network, ID\_Programmer, Details)

POSTARE (#ID\_Post, ID\_Account, Titlu, Text, Data)

COMENTARIU (#ID\_Comentariu, ID\_Account, Text, Data)

LOCAȚIE (#ID\_Location, Stat, Oraș)

INTERVENȚIE (#ID\_Intervention, ID\_Moderator, ID\_Post, ID\_Comentariu, Detalii, Rezultat)

MODERATOR (#ID\_Moderator, ID\_Location, Nume, Prenume)

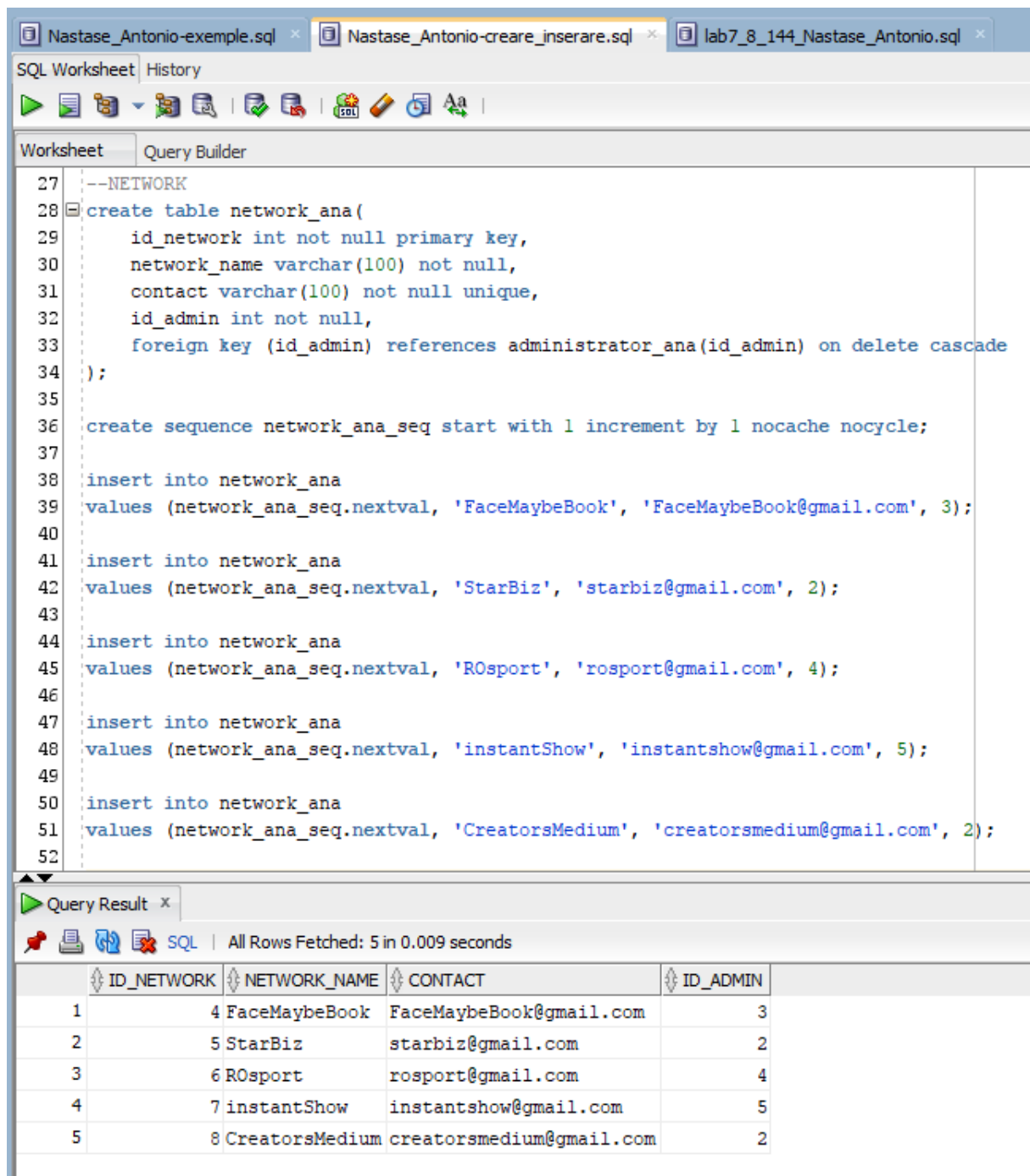
9. Realizarea normalizării până la forma normală 3 (FN1-FN3).

Presupunem că în tabelul UTILIZATOR există un atribut care reține toți prietenii unui utilizator. Acesta nu se află în FN1, deoarece valoarea atributului anterior menționat nu este indivizibilă. Pentru a evita o astfel de valoare, alegem să adăugăm o relație recursivă la tabelul UTILIZATOR.

Presupunem că tabelul POSTARE este de forma POSTARE (ID\_Postare#, ID\_Comentariu#, ID\_Account, Titlu, Text\_Postare, Text\_Comentariu, Data\_postare, Data\_comentariu). Acesta nu se află în FN2, deoarece Text\_comentariu și Data\_comentariu depind doar de ID\_Comentariu. O soluție este împărțirea în 2 tabele POSTARE și COMENTARIU.

Presupunem că tabelul UTILIZATOR este de forma UTILIZATOR (#ID\_User, ID\_Friend, ID\_Location, Nume, Prenume, State, City). Acesta nu se află în FN3, deoarece attributele State și City, care nu sunt chei, depind de ID\_Location, care nu este cheie primară. O soluție este crearea unui nou tabel LOCAȚIE (#ID\_Location, State, City).

10. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).



```
27 --NETWORK
28 create table network_ana(
29     id_network int not null primary key,
30     network_name varchar(100) not null,
31     contact varchar(100) not null unique,
32     id_admin int not null,
33     foreign key (id_admin) references administrator_ana(id_admin) on delete cascade
34 );
35
36 create sequence network_ana_seq start with 1 increment by 1 nocache nocycle;
37
38 insert into network_ana
39 values (network_ana_seq.nextval, 'FaceMaybeBook', 'FaceMaybeBook@gmail.com', 3);
40
41 insert into network_ana
42 values (network_ana_seq.nextval, 'StarBiz', 'starbiz@gmail.com', 2);
43
44 insert into network_ana
45 values (network_ana_seq.nextval, 'ROsport', 'rospport@gmail.com', 4);
46
47 insert into network_ana
48 values (network_ana_seq.nextval, 'instantShow', 'instantshow@gmail.com', 5);
49
50 insert into network_ana
51 values (network_ana_seq.nextval, 'CreatorsMedium', 'creatorsmedium@gmail.com', 2);
52
```

Query Result x

All Rows Fetched: 5 in 0.009 seconds

	ID_NETWORK	NETWORK_NAME	CONTACT	ID_ADMIN
1	4	FaceMaybeBook	FaceMaybeBook@gmail.com	3
2	5	StarBiz	starbiz@gmail.com	2
3	6	ROsport	rospport@gmail.com	4
4	7	instantShow	instantshow@gmail.com	5
5	8	CreatorsMedium	creatorsmedium@gmail.com	2

Nastase\_Antonio-exemple.sql × Nastase\_Antonio-creare\_inserare.sql × lab7\_8\_144\_Nastase\_Antonio.sql ×

SQL Worksheet History

Worksheet Query Builder

```

132 --ACCOUNT
133 create table account_ana(
134     id_account int not null primary key,
135     username varchar(100) not null,
136     email varchar(100) not null unique,
137     id_network int not null,
138     id_user int not null,
139     foreign key (id_network) references network_ana(id_network) on delete cascade,
140     foreign key (id_user) references user_ana(id_user) on delete cascade
141 );
142
143 create sequence account_ana_seq start with 1 increment by 1 nocache nocycle;
144
145 insert into account_ana
146 values (account_ana_seq.nextval, 'ruben1', 'ruben1@gmail.com', 6, 5);
147
148 insert into account_ana
149 values (account_ana_seq.nextval, 'frenzy2', 'frenzy2@gmail.com', 4, 2);
150
151 insert into account_ana
152 values (account_ana_seq.nextval, 'olof3', 'olof3@gmail.com', 8, 6);

```

Query Result ×

SQL | All Rows Fetched: 10 in 0.01 seconds

	ID_ACCOUNT	USERNAME	EMAIL	ID_NETWORK	ID_USER
1	18	ioana	ioana@gmail.com	8	4
2	20	ruben1	ruben1@gmail.com	6	5
3	21	frenzy2	frenzy2@gmail.com	4	2
4	22	olof3	olof3@gmail.com	8	6
5	23	carlos4	carlos4@gmail.com	8	4
6	24	benny5	benny5@gmail.com	6	3
7	25	cristiano6	cristiano6@gmail.com	5	2
8	26	benito7	benito7@gmail.com	7	6
9	27	alvaro8	alvaro8@gmail.com	7	6
10	28	scott9	scott9@gmail.com	8	3



## 11. Formulați în limbaj natural și implementați 5 cereri SQL complexe.

The image displays three screenshots of an SQL Worksheet interface, showing the execution of complex SQL queries and their results.

**First Screenshot:** The SQL query is as follows:

```
16
17
18 --Sa se afiseze orasul unde au avut loc cele mai multe interventii asupra postarilor
19 select v_main.city
20 from(
21     select v.city city
22     from (select count(i.id_intervention) numar, l.city city from location_ana l
23
24
25
26     order by v.numar
27 ) v_main
28 where rownum < 2;
```

The query result shows:

CITY
1 Lyon

**Second Screenshot:** The SQL query is as follows:

```
32 --Sa se afiseze cele mai recente postari pentru utilizatorul 3 si cele mai recente comentarii pentru 6.
33 --Daca utilizatorul nu are nicio postare/comentariu, se va afisa 'Nu exista'
34 with comment_table as (select c.comment_date, c.id_account, a.id_user, c.comment_text from comment_ana c join account_ana a on c.id_account = a.id_account)
35 select decode(u.id_user,
36 3, nvl((select p.post_text from post_ana p join account_ana a on p.id_account = a.id_account where a.id_user = u.id_user
37 and p.post_date = (select max(post_date) from post_ana p join account_ana a on p.id_account = a.id_account where a.id_user = u.id_user)), 'Nu exista'),
38 6, nvl((select a.comment_text from comment_table a where a.id_user = u.id_user
39 and a.comment_date = (select max(comment_date) from comment_table a where a.id_user = u.id_user))), 'Nu exista')
40 ) as "Continut"
41 from user_ana u
42 where u.id_user = 3 or u.id_user = 6;
```

The query result shows:

Continut
1 Alerg
2 Cam asa

**Third Screenshot:** The SQL query is as follows:

```
45 --Sa se afiseze numele complet al fiecarui utilizator care a fost tinta la mai mult de o interventie
46 select u.last_name || ' ' || u.first_name as "Nume complet"
47 from intervention_ana i join comment_ana c on c.id_post = i.id_comment
48     join account_ana a on a.id_account = c.id_account
49     join user_ana u on u.id_user = a.id_user
50 group by u.last_name, u.first_name
51 having count(id_intervention) > 1;
```

The query result shows:

Nume complet
1 Borris Arnold
2 Robertson Bill

Nastase\_Antonio-exemple.sql × Nastase\_Antonio-creare\_inserare.sql × lab7\_8\_144\_Nastase\_Antonio.sql ×

SQL Worksheet History

Worksheet Query Builder

```

10
11
12 --Sa se afiseze de cate luni nu s-a mai postat pe retea ce contine 'Face' in denumirea sa.
13 select distinct round(months_between(sysdate, p.post_date), 2) as "Numar luni"
14 from post_ana p join account_ana a on p.id_account = a.id_account join network_ana n on n.id_network = a.id_network
15 where instr(upper(n.network_name), 'FACE') <> 0;
16

```

Script Output × Query Result ×

SQL All Rows Fetched: 1 in 0.009 seconds

	Numar luni
1	0.04

Nastase\_Antonio-exemple.sql × Nastase\_Antonio-creare\_inserare.sql × lab7\_8\_144\_Nastase\_Antonio.sql ×

SQL Worksheet History

Worksheet Query Builder

```

1 --11
2
3 --Sa se afiseze toti moderatorii care au intervenit la comentariile utilizatorului cu numele de familie Robertson
4 select m.first_name, m.last_name
5 from moderator_ana m join intervention_ana i on m.id_moderator = i.id_moderator
6      join comment_ana c on c.id_post = i.id_comment
7      join account_ana a on a.id_account = c.id_account
8      join user_ana u on u.id_user = a.id_user
9 where lower(u.last_name) = 'robertson';
10

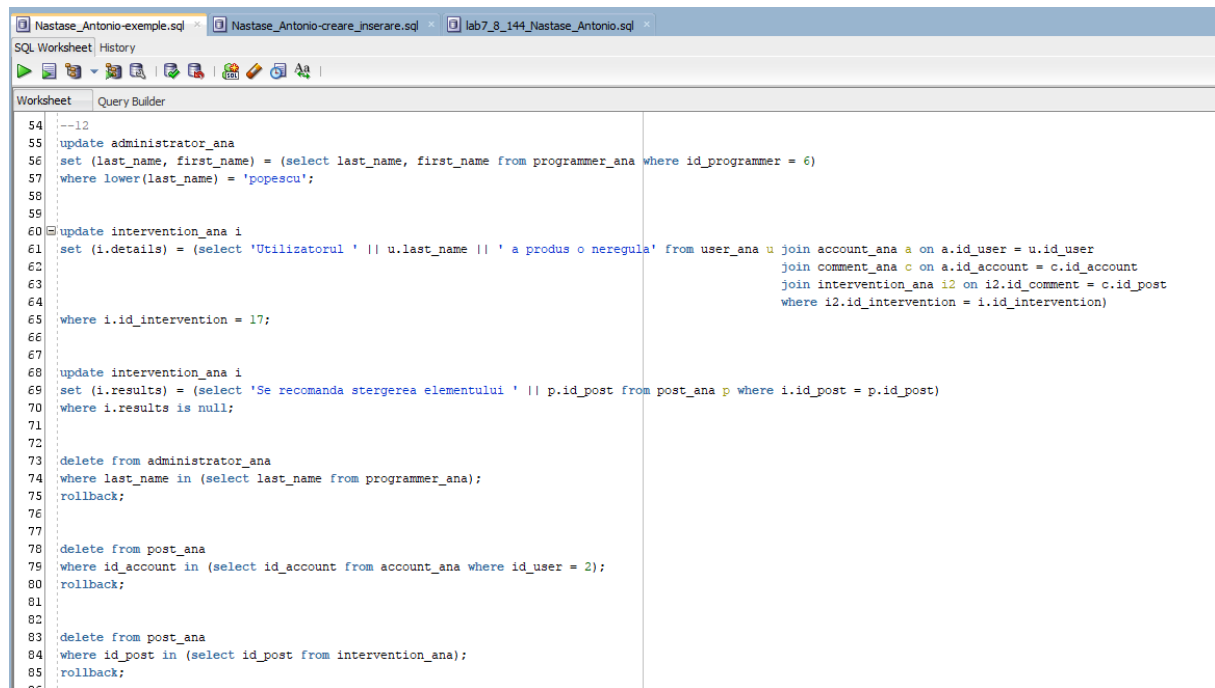
```

Script Output × Query Result ×

SQL All Rows Fetched: 2 in 0.011 seconds

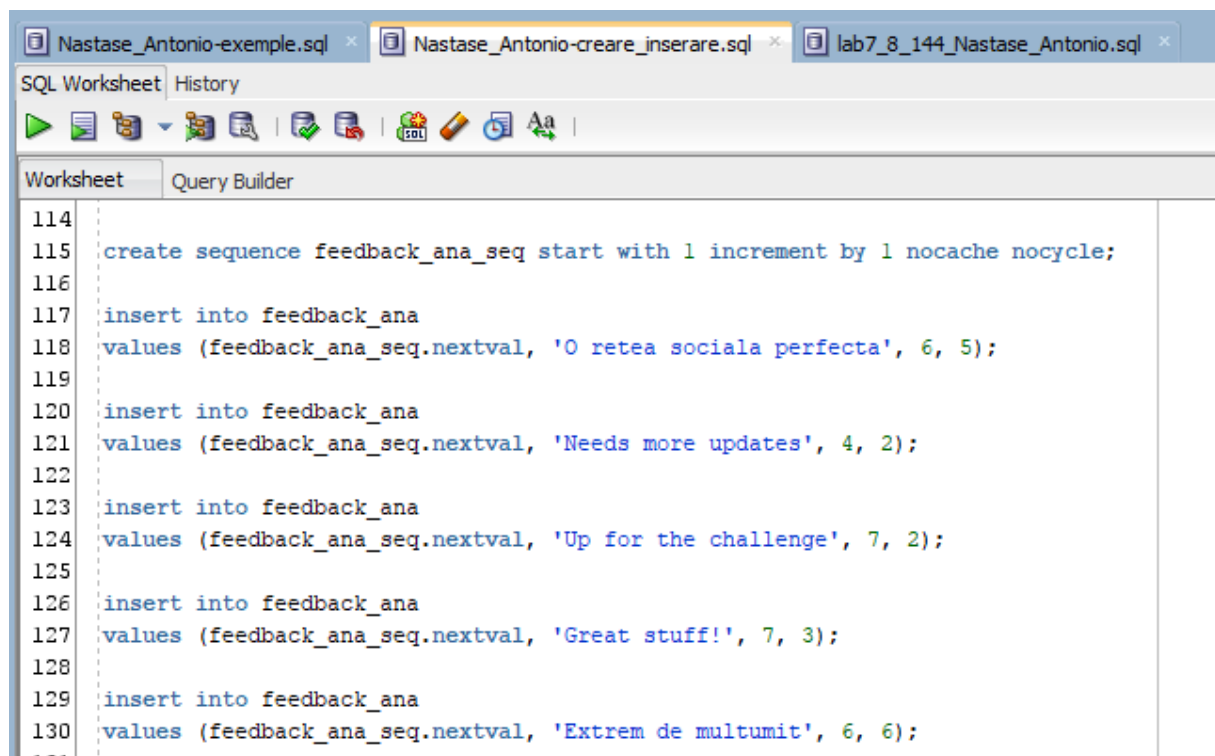
	FIRST_NAME	LAST_NAME
1	Cristian	Stan
2	Inna	James

12. Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri.



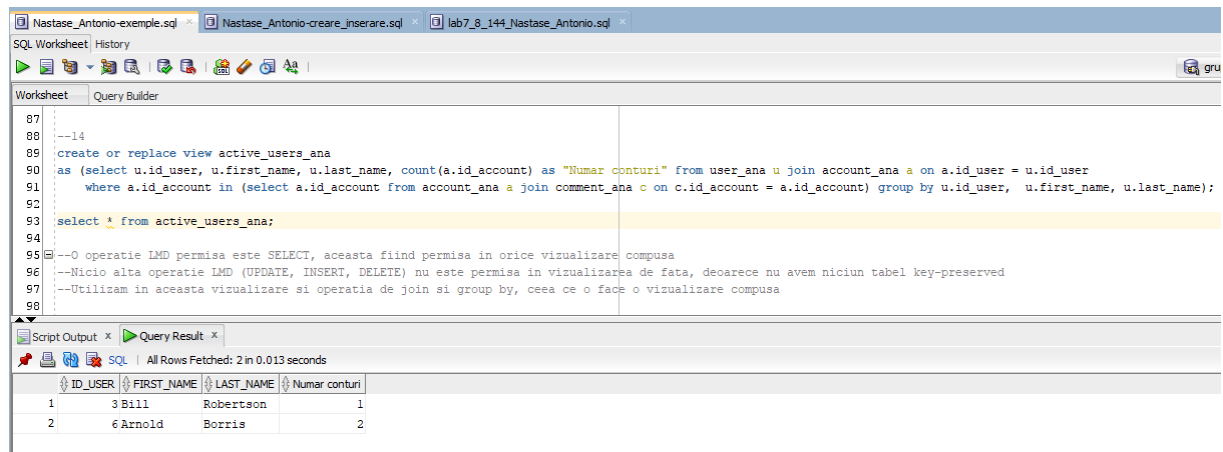
```
--12
54 update administrator_ana
55 set (last_name, first_name) = (select last_name, first_name from programmer_ana where id_programmer = 6)
56 where lower(last_name) = 'popescu';
57
58
59
60 update intervention_ana i
61 set (i.details) = (select 'Utilizatorul ' || u.last_name || ' a produs o neregula' from user_ana u join account_ana a on a.id_user = u.id_user
62 join comment_ana c on a.id_account = c.id_account
63 join intervention_ana i2 on i2.id_comment = c.id_post
64 where i2.id_intervention = i.id_intervention)
65 where i.id_intervention = 17;
66
67
68 update intervention_ana i
69 set (i.results) = (select 'Se recomanda stergerea elementului ' || p.id_post from post_ana p where i.id_post = p.id_post)
70 where i.results is null;
71
72
73 delete from administrator_ana
74 where last_name in (select last_name from programmer_ana);
75 rollback;
76
77
78 delete from post_ana
79 where id_account in (select id_account from account_ana where id_user = 2);
80 rollback;
81
82
83 delete from post_ana
84 where id_post in (select id_post from intervention_ana);
85 rollback;
86
```

13. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 10).



```
114
115 create sequence feedback_ana_seq start with 1 increment by 1 nocache nocycle;
116
117 insert into feedback_ana
118 values (feedback_ana_seq.nextval, 'O retea sociala perfecta', 6, 5);
119
120 insert into feedback_ana
121 values (feedback_ana_seq.nextval, 'Needs more updates', 4, 2);
122
123 insert into feedback_ana
124 values (feedback_ana_seq.nextval, 'Up for the challenge', 7, 2);
125
126 insert into feedback_ana
127 values (feedback_ana_seq.nextval, 'Great stuff!', 7, 3);
128
129 insert into feedback_ana
130 values (feedback_ana_seq.nextval, 'Extrem de multumit', 6, 6);
131
```

14. Crearea unei vizualizări compuse. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.



The screenshot shows a SQL Worksheet with the following SQL code:

```
--14
create or replace view active_users_ana
as (select u.id_user, u.first_name, u.last_name, count(a.id_account) as "Numar conturi" from user_ana u join account_ana a on a.id_user = u.id_user
where a.id_account in (select a.id_account from account_ana a join comment_ana c on c.id_account = a.id_account) group by u.id_user, u.first_name, u.last_name);

select * from active_users_ana;
```

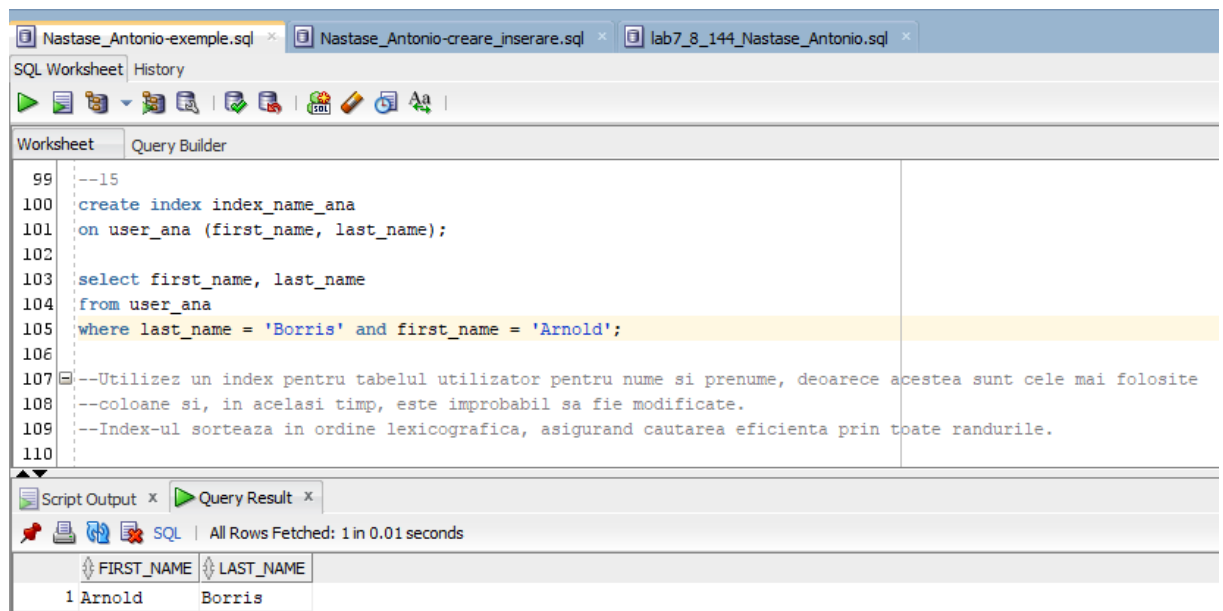
Below the code, there are comments in Romanian explaining LMD (Least Modification) permissions:

- O operatie LMD permisa este SELECT, aceasta fiind permisa in orice vizualizare compusa
- Nicio alta operatie LMD (UPDATE, INSERT, DELETE) nu este permisa in vizualizarea de fata, deoarece nu avem niciun tabel key-preserved
- Utilizam in aceasta vizualizare si operatia de join si group by, ceea ce o face o vizualizare compusa

The Query Result shows the following data:

ID_USER	FIRST_NAME	LAST_NAME	Numar conturi
3	Bill	Robertson	1
6	Arnold	Borris	2

15. Crearea unui index care să optimizeze o cerere de tip căutare cu 2 criterii. Specificați cererea.



The screenshot shows a SQL Worksheet with the following SQL code:

```
--15
create index index_name_ana
on user_ana (first_name, last_name);

select first_name, last_name
from user_ana
where last_name = 'Borris' and first_name = 'Arnold';
```

Below the code, there are comments in Romanian explaining the index:

- Utilizez un index pentru tabelul utilizator pentru nume si prenume, deoarece acestea sunt cele mai folosite
- coloane si, in acelasi timp, este improbabil sa fie modificate.
- Index-ul sorteaza in ordine lexicografica, asigurand cautarea eficienta prin toate randurile.

The Query Result shows the following data:

FIRST_NAME	LAST_NAME
1 Arnold	Borris

16. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minimum 4 tabele și două cereri ce utilizează operația division.

Nastase\_Antonio-exemple.sql x Nastase\_Antonio-creare\_inserare.sql x lab7\_8\_144\_Nastase\_Antonio.sql x

SQL Worksheet History

Worksheet Query Builder

```

111 --16
112
113 --Pentru fiecare retea vreau sa aflu numele administratorului, daca are sau nu actualizari, daca are sau nu feedback-uri si numarul de conturi
114 select n.network_name, ad.last_name, decode(u.id_network, null, 'Nu', 'Da') as "Este actualizata",
115        decode(f.id_network, null, 'Nu', 'Da') as "Are feedback", count(a.id_account)
116 from network_ana n left outer join administrator_ana ad on ad.id_admin = n.id_admin
117        left outer join feedback_ana f on n.id_network = f.id_network
118        left outer join account_ana a on a.id_network = n.id_network
119        left outer join update_ana u on u.id_network = n.id_network
120 group by n.id_network, n.network_name, ad.last_name, f.id_network, u.id_network;
121
122

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.012 seconds

	NETWORK_NAME	LAST_NAME	Este actualizata	Are feedback	COUNT(A.ID_ACCOUNT)
1	StarBiz	Leclerc	Da	Nu	1
2	ROsport	Tommy	Da	Da	16
3	FaceMaybeBook	Popescu	Da	Da	3
4	instantShow	Robert	Da	Da	4
5	CreatorsMedium	Leclerc	Da	Nu	4

Nastase\_Antonio-exemple.sql x Nastase\_Antonio-creare\_inserare.sql x

SQL Worksheet History

Worksheet Query Builder

```

group by n.id_network, n.network_name, ad.last_name, f.id_network, u.id_network;

--Afisati toate titlurile de postari care au acelasi text si data cu un comentariu.

update comment_ana
set comment_text = (select post_text from post_ana where id_post = 11), comment_date = (select post_date from post_ana where id_post = 11)
where id_post = 3;

select title
from post_ana
where (post_text, post_date) in (select comment_text, comment_date from comment_ana);

```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.008 seconds

	TITLE
1	Despre sport

Nastase\_Antonio-exemple.sql x Nastase\_Antonio-creare\_inserare.sql x

SQL Worksheet History

Worksheet Query Builder

```

--Afisati datele de contact ale administratorilor care sunt si programatori.

update administrator_ana
set last_name = (select last_name from programmer_ana where id_programmer = 4), first_name = (select first_name from programmer_ana where id_programmer = 4)
where id_admin = 5;

select contact
from administrator_ana
where (last_name, first_name) in (select last_name, first_name from programmer_ana);

```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.009 seconds

	CONTACT
1	06333293

17. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebrică, arbore algebric și limbaj (SQL), atât anterior cât și ulterior optimizării.

--Sa se afiseze date despre toate rețelele care au primit un update pe partea de frontend, detaliile actualizării si numele programatorului.

Cod SQL:

```
select n.network_name, n.contact, u.details, p.last_name
from network_ana n join update_ana u using (id_network)
                join programmer_ana p using (id_programmer)
where lower(p.job_title) = 'frontend';
```

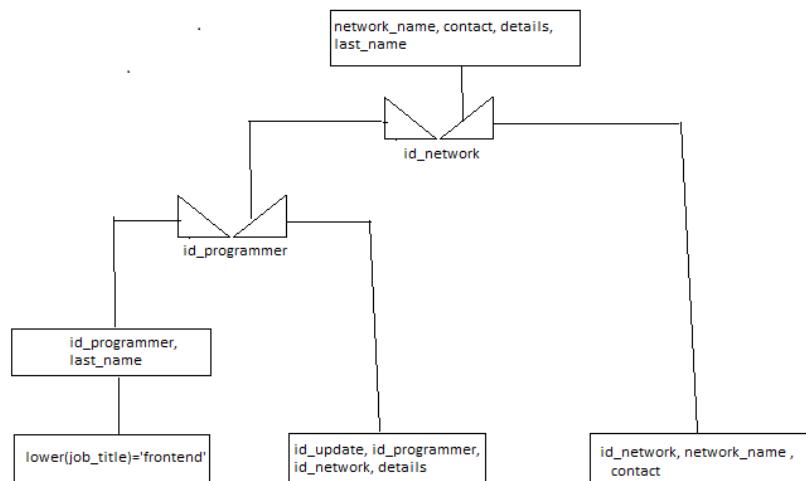
Cererea optima exprimata prin expresii algebrice:

```
R1 = SELECT(PROGRAMMER_ANA, lower(job_title) = 'frontend')
R2 = PROJECT(R1, id_programmer, last_name)
R3 = PROJECT(UPDATE_ANA, id_update, id_programmer, id_network, details)
R4 = PROJECT(NETWORK_ANA, id_network, network_name, contact)
R5 = JOIN(R2, R3, id_programmer)
R6 = JOIN(R4, R5, id_network)
R7 = PROJECT(R6, network_name, contact, details, last_name)
```

Justificare:

Cererea este optima, fiindca operatiile de select se executa cat mai rapid si operatiile de project se executa printre primele, asigurandu-ne in acest fel ca inlaturam atributele care nu ne sunt utile. De asemenea, inainte de operatiile de join folosim proiectia, pentru a nu se aplica operatia de join pe toate coloanele din table.

Arborele algebric pentru cererea optimizata:



18. a. Realizarea normalizării BCNF, FN4, FN5.

Presupunem că în tabelul ADMINISTRATOR, atributul contact determină funcțional cheia primară id\_admin. Acesta nu se mai află în BNFC, deoarece atributul contact este unic și determină cheia primară. Pentru a rezolva problema, vom impune ca id\_admin să fie ales aleator.

Presupunem că în tabelul UPDATE, cheia primară id\_update este determinată de id\_programator și id\_network, anume fiecare programator ar putea face un singur update într-o rețea. Acest fapt încalcă restricțiile din FN5, deoarece se creează dependențe multiple (id\_update – id\_programator și id\_update – id\_network). Vom alege id\_update aleator.

Presupunem că între tabelele ADMINISTRATOR și PROGRAMATOR ar exista o relație de coordonare (one-to-many obligatorie). Acest fapt ar rezulta într-o dependență ciclică (administrator - programator – update – rețea - administrator), ceea ce ar încălca regulile specifice FN5. Eliminăm această relație, deoarece o putem obține folosind operația de join.

b. Aplicarea denormalizării, justificând necesitatea acesteia.

Adaug o relație one-to-many între INTERVENȚIE și UTILIZATOR. Aceasta creează o dependență ciclică și anulează apartenența la FN5. Denormalizarea este utilă în acest caz, deoarece erau necesare 3 operații de join pentru a afla care este utilizatorul afectat de o anumită intervenție.