

Baza de Date: MySocial

Stancu Rares: Grupa 144

May 2024

Contents

1	Descrierea modelului real, a utilității acestuia și a regulilor de funcționare	4
1.1	Tema	4
1.2	Descrierea modelului real	4
1.3	Utilitatea	4
1.4	Funcționalități	4
2	Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului	5
2.1	Tabela Utilizator	5
2.2	Tabela Postari	5
2.3	Tabela Video	5
2.4	Tabela Like	6
2.5	Tabela Comentarii	6
2.6	Tabela Reactie	6
2.7	Tabela Tag	6
2.8	Tabela Follow	7
2.9	Tabela Poze	7
2.10	Tabela Participant_Eventiment	7
2.11	Tabela Membru_Grup	7
2.12	Tabela Grup	7
2.13	Tabela Eveniment	8
2.14	Tabela Mesaj	8
3	Descrierea entităților, incluzând precizarea cheii primare.	8
3.1	Entitatea Utilizator	8
3.2	Entitatea Postari	8
3.3	Entitatea Video	8
3.4	Entitatea Like	9
3.5	Entitatea Comentarii	9
3.6	Entitatea Reactie	9
3.7	Entitatea Tag	9
3.8	Entitatea Follow	9
3.9	Entitatea Poze	9
3.10	Entitatea Participant_Eventiment	9

3.11	Entitatea Membru_Grup	10
3.12	Entitatea Grup	10
3.13	Entitatea Eveniment	10
3.14	Entitatea Mesaj	10
4	Descrierea relațiilor, incluzând precizarea cardinalității acestora.	10
4.0.1	Tabela Utilizator	10
4.0.2	Tabela Postari	11
4.0.3	Tabela Video	11
4.0.4	Tabela Like	11
4.0.5	Tabela Comentarii	12
4.0.6	Tabela Reactie	12
4.0.7	Tabela Tag	12
4.0.8	Tabela Follow	12
4.0.9	Tabela Poze	12
4.0.10	Tabela ParticipantEveniment	12
4.0.11	Tabela MembruGrup	12
4.0.12	Tabela Grup	13
4.0.13	Tabela Eveniment	13
4.0.14	Tabela Mesaj	13
5	Descrierea atributelor, incluzand tipul de date și eventualele constrangeri, valori implicite, valori posibile ale atributelor	13
5.0.1	Tabela Utilizator	13
6	Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5	19
7	Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6	20
8	Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7	20
9	Normalizarea tabelelor până la forma normală 3 (FN1-FN3)	21
10	Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).	23
11	Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea	29
11.1	Pentru Tabela Utilizator	29
11.2	Pentru Tabela Postari	31
11.3	Pentru Tabela Video	35
11.4	Pentru Tabela Like	37
11.5	Pentru Tabela Comentarii	40
11.6	Pentru Tabela Reactie	42
11.7	Pentru Tabela Tag	46

11.8	Pentru Tabela Follow	49
11.9	Pentru Tabela Poze	52
11.10	Pentru Tabela Grup	55
11.11	Pentru Tabela Membru _{Grup}	58
11.12	Pentru Tabela Eveniment	61
11.13	Pentru Tabela Participant _{Eveniment}	64
11.14	Pentru Tabela Mesaj	66
12	Formulați în limbaj natural și implementați 5 cereri SQL complexe	69
13	Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.	82
13.1	Actualizarea coloanei "TIP_REACTIE" din tabela Reactie pentru utilizatorii care au reactionat cu "Wow" in "Omg".	82
13.2	Stergerea înregistrărilor din tabela Postari pentru comentariile care se referă la postări mai vechi de 2023	84
13.3	Ștergere: Eliminarea comentariilor postate înainte de anul 2022 pentru postările care au primit reacții de tip "Omg"	86
13.4	Actualizare: Schimbarea parolelor utilizatorilor care au creat postări video mai scurte de 40 minute.	88

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

1.1 Tema

Tema acestui proiect este reprezentată de către o bază de date realizată pentru o aplicație numită "MySocial", care are ca scop stocarea și gestionarea datelor utilizatorilor aplicației.

1.2 Descrierea modelului real

Această Bază de date poate fi utilizată pentru a reține postările utilizatorilor, like-urile unei postări și alte funcționalități ale aplicației.

Entitatea "Utilizator" conține informații despre utilizatori, cum ar fi ID-ul utilizatorului, numele, email-ul și data înregistrării acestuia. Entitatea Postări conține informații despre o anumită postare a unui utilizator, precum un id unic, data postării, id-ul utilizatorului. Această entitate este legată de entitățile like, video, comentarii dar și alte entități similare ce conțin informații relevante despre postare cum ar fi utilizatorii care au dat like la postare, comentariile postării și numele utilizatorilor ce au dat comentarii. O altă entitate relevantă este "Eveniment" ce deține informații despre numele, data evenimentului, numele participanților, dar și alte date importante.

1.3 Utilitatea

Această bază de date pentru o platformă de socializare are multiple utilități precum:

- Gestionarea conținutului: Permite utilizatorilor să creeze și să gestioneze postări, comentarii și like-uri asociate acestora. Acest lucru ajută la organizarea și prezentarea conținutului generat de utilizatori într-un mod eficient.
- Comunicare privată: Posibilitatea de a trimite mesaje între utilizatori oferă o modalitate privată și directă de comunicare, care poate fi folosită pentru conversații personale, planificarea evenimentelor sau colaborarea în grupuri.
- Organizarea și managementul comunităților: Permite utilizatorilor să creeze și să gestioneze pagini și grupuri în care pot aduna persoane cu interese comune. Aceste pagini și grupuri pot fi utilizate pentru a împărtăși informații, pentru a organiza evenimente sau pentru a facilita discuțiile într-un cadru specific.
- Analiza comportamentului utilizatorilor: Datele colectate în baza de date pot fi folosite pentru analiza comportamentului utilizatorilor și pentru înțelegerea preferințelor și nevoilor lor. Aceste informații pot fi folosite pentru a optimiza caracteristicile platformei și pentru a lua decizii informate în ceea ce privește dezvoltarea și îmbunătățirea acesteia.

1.4 Funcționalități

- Interacțiunea cu postările: Utilizatorii pot interacționa cu postările prin acordarea de like-uri și lăsarea de comentarii. Fiecare utilizator poate da like la mai multe postări și poate lăsa mai multe comentarii la diferite postări.

- Participarea în grupuri: Utilizatorii pot fi membri în mai multe grupuri și pot interacționa cu alți membri ale acestor grupuri prin postări, comentarii și like-uri.

Prin urmare, baza de date propusă pentru platforma de socializare oferă o gamă variată de funcționalități care contribuie la crearea unei experiențe captivante și interactive pentru utilizatori. Prin gestionarea postărilor, interacțiunile sociale, comunicarea privată și gestionarea comunităților online, această bază de date facilitează conexiunile între utilizatori și promovează angajamentul și interacțiunea activă pe platformă.

2 Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului

2.1 Tabela Utilizator

- **id_utilizator:** Cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- **nume:** Trebuie să aibă o lungime maximă de 40 de caractere ce reprezintă un nume.
- **email:** Trebuie să aibă o lungime maximă de 100 de caractere ce reprezintă un email.
- **parola:** Trebuie să aibă o lungime maximă de 100 de caractere și minim de 1.
- **data_inregistrare:** Trebuie să fie o valoare de tipul data.

2.2 Tabela Postari

- **id_postari:** Cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- **id_utilizator:** este o cheie externă către tabela Utilizator și trebuie să se refere la un id_utilizator existent în acea tabelă.
- **data_postarii:** Trebuie să fie o valoare de tipul data.

2.3 Tabela Video

- **id_video:** Cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- **id_postari:** este o cheie externă către tabela Postari și trebuie să se refere la un id_postari existent în acea tabelă.
- **titlu:** Trebuie să aibă o lungime maximă de 255 de caractere ce reprezintă un titlu de videoclip.
- **durata:** Trebuie să aibă o valoare numerică pozitivă mai mare ca 0.
- **data_upload:** Trebuie să fie o valoare de tipul data.
- **url:** Trebuie să aibă o lungime maximă de 255 de caractere ce reprezintă linkul videoclipului

2.4 Tabela Like

- **id_like:** Cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- **id_utilizator:** este o cheie externă către tabela Utilizator și trebuie să se refere la un id_utilizator existent în acea tabelă.
- **id_postari:** este o cheie externă către tabela Postari și trebuie să se refere la un id_postari existent în acea tabelă.
- **data_like:** Trebuie să fie o valoare de tipul data care reprezintă data în care a fost dat like-ul.

2.5 Tabela Comentarii

- **id_comentarii:** Cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- **id_utilizator:** este o cheie externă către tabela Utilizator și trebuie să se refere la un id_utilizator existent în acea tabelă.
- **id_postari:** este o cheie externă către tabela Postari și trebuie să se refere la un id_postari existent în acea tabelă.
- **data_comentariu:** Trebuie să fie o valoare de tipul data care reprezintă data în care a fost dat comentariul.
- **continut:** Trebuie să aibă o lungime maximă de 255 de caractere și o lungime minimă de 1 caracter.

2.6 Tabela Reactie

- **id_Reactie:** Cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- **id_postari:** este o cheie externă către tabela Postari și trebuie să se refere la un id_postari existent în acea tabelă.
- **tip_reactie:** Trebuie să aibă o lungime maximă de 25 de caractere și o lungime minimă de 1 caracter care să reprezinte o reacție la postare.
- **data_reactie:** Trebuie să fie o valoare de tipul data care reprezintă data în care a fost data reacția.

2.7 Tabela Tag

- **id_tag:** Cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- **id_postari:** este o cheie externă către tabela Postari și trebuie să se refere la un id_postari existent în acea tabelă.
- **nume_tag:** Trebuie să aibă o lungime maximă de 100 de caractere și o lungime minimă de 1 caracter care să reprezinte tag-ul unei postări.

2.8 Tabela Follow

- **id_follow:** Cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- **id_utilizator:** este o cheie externă către tabela Utilizator și trebuie să se refere la un id_utilizator existent în acea tabelă.
- **data_follow:** Trebuie să fie o valoare de tipul data, ce reprezintă data în care a dat follow-ul

2.9 Tabela Poze

- **id_poze:** Cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- **id_postari:** este o cheie externă către tabela Postari și trebuie să se refere la un id_postari existent în acea tabelă.
- **nume_poza:** Trebuie să aibă o lungime maximă de 100 de caractere și o lungime minimă de 1 caracter care să reprezinte numele unei poze.
- **data_upload:** Trebuie să fie o valoare de tipul data care reprezintă data în care a fost postată poza.

2.10 Tabela Participant_Eventiment

- **id_participant_eventiment:** Cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- **id_eventiment:** este o cheie externă către tabela Eveniment și trebuie să se refere la un id_eventiment existent în acea tabelă.
- **id_utilizator:** este o cheie externă către tabela Utilizator și trebuie să se refere la un id_utilizator existent în acea tabelă.
- **data_inregistrare:** Trebuie să fie o valoare de tipul data.

2.11 Tabela Membru_Grup

- **id_membru_grup:** Cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- **id_grup:** este o cheie externă către tabela Grup și trebuie să se refere la un id_grup existent în acea tabelă.
- **id_utilizator:** este o cheie externă către tabela Utilizator și trebuie să se refere la un id_utilizator existent în acea tabelă.
- **data_intrare:** Trebuie să fie o valoare de tipul data.

2.12 Tabela Grup

- **id_grup:** Cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- **Nume:** Trebuie să aibă o lungime maximă de 100 de caractere ce reprezintă numele grupului.

2.13 Tabela Eveniment

- **id_eveniment:** Cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- **Nume:** Trebuie să aibă o lungime maximă de 255 de caractere ce reprezintă numele evenimentului.
- **data_eveniment:** Trebuie să fie o valoare de tipul data.

2.14 Tabela Mesaj

- **id_mesaj:** Cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- **id_utilizator_sender:** este o cheie externă către tabela Utilizator și trebuie să se refere la un id_utilizator existent în acea tabelă care a trimis mesajul.
- **id_utilizator_receiver:** este o cheie externă către tabela Utilizator și trebuie să se refere la un id_utilizator existent în acea tabelă care a primit mesajul.
- **continut:** Trebuie să aibă o lungime maximă de 500 de caractere ce reprezintă un mesaj trimis de un utilizator
- **data_trimitere:** Trebuie să fie o valoare de tipul data.

3 Descrierea entităților, incluzând precizarea cheii primare.

3.1 Entitatea Utilizator

Această tabelă conține informații despre utilizatorii platformei. Fiecare înregistrare este identificată prin cheia primară `id_utilizator` și conține informații despre nume (`nume`), adresa de email (`email`), parola (`parola`) și data înregistrării (`data_inregistrare`). Aceste informații sunt esențiale pentru gestionarea conturilor utilizatorilor și pentru autentificarea și comunicarea în cadrul platformei.

3.2 Entitatea Postari

Această tabelă conține informații despre postările făcute de utilizatori. Fiecare înregistrare este identificată prin cheia primară `id_postari` și include referințe către utilizator (`id_utilizator`) și data postării (`data_postarii`). Această structură permite asocierea postărilor cu conținutul și interacțiunile relevante.

3.3 Entitatea Video

Această tabelă conține informații despre videoclipurile încărcate de utilizatori. Fiecare înregistrare este identificată prin cheia primară `id_video` și include referințe către postare (`id_postari`), titlul videoclipului (`titlu`), durata (`durata`), data upload-ului (`data_upload`) și URL-ul videoclipului (`url`). Aceste informații sunt utilizate pentru gestionarea și afișarea videoclipurilor în cadrul postărilor.

3.4 Entitatea Like

Această tabelă conține informații despre like-urile date de utilizatori la postări. Fiecare înregistrare este identificată prin cheia primară `id_like` și include referințe către utilizator (`id_utilizator`), postare (`id_postari`) și data like-ului (`data_like`). Aceste informații sunt importante pentru urmărirea interacțiunilor utilizatorilor cu conținutul postat.

3.5 Entitatea Comentarii

Această tabelă conține informații despre comentariile făcute de utilizatori la postări. Fiecare înregistrare este identificată prin cheia primară `id_comentarii` și include referințe către utilizator (`id_utilizator`), postare (`id_postari`), data comentariului (`data_comentariu`) și conținutul comentariului (`continut`). Aceste informații sunt cruciale pentru facilitarea comunicării și feedback-ului între utilizatori.

3.6 Entitatea Reactie

Această tabelă conține informații despre reacțiile utilizatorilor la postări. Fiecare înregistrare este identificată prin cheia primară `id_reactie` și include referințe către postare (`id_postari`), tipul reacției (`tip_reactie`) și data reacției (`data_reactie`). Aceste informații ajută la urmărirea sentimentului utilizatorilor față de conținutul postat.

3.7 Entitatea Tag

Această tabelă conține informații despre tag-urile asociate cu postările. Fiecare înregistrare este identificată prin cheia primară `id_tag` și include referințe către postare (`id_postari`) și numele tag-ului (`nume_tag`). Aceste informații sunt utile pentru organizarea și căutarea postărilor după categorii sau subiecte.

3.8 Entitatea Follow

Această tabelă conține informații despre relațiile de follow dintre utilizatori. Fiecare înregistrare este identificată prin cheia primară `id_follow` și include referințe către utilizator (`id_utilizator`) și data follow-ului (`data_follow`). Aceste informații sunt esențiale pentru gestionarea rețelei sociale și a interacțiunilor dintre utilizatori.

3.9 Entitatea Poze

Această tabelă conține informații despre pozele încărcate de utilizatori. Fiecare înregistrare este identificată prin cheia primară `id_poze` și include referințe către postare (`id_postari`), numele pozei (`nume_poze`) și data upload-ului (`data_upload`). Aceste informații sunt utilizate pentru gestionarea și afișarea pozelor în cadrul postărilor.

3.10 Entitatea Participant_Eventiment

Această tabelă conține informații despre utilizatorii care participă la evenimente. Fiecare înregistrare este identificată prin cheia primară `id_participant_eventiment` și include referințe către utilizator

(id_utilizator), eveniment (id_eveniment) și data înregistrării (data_inregistrare). Aceste informații sunt importante pentru organizarea și gestionarea participării la evenimente.

3.11 Entitatea Membru_Grup

Această tabelă conține informații despre utilizatorii care fac parte din grupuri. Fiecare înregistrare este identificată prin cheia primară id_membru_grup și include referințe către grup (id_grup), utilizator (id_utilizator) și data intrării în grup (data_intrare). Aceste informații sunt esențiale pentru gestionarea membrilor grupurilor și a activităților acestora.

3.12 Entitatea Grup

Această tabelă conține informații despre grupurile create în cadrul platformei. Fiecare înregistrare este identificată prin cheia primară id_grup și include numele grupului (nume). Aceste informații sunt necesare pentru organizarea și gestionarea grupurilor de utilizatori.

3.13 Entitatea Eveniment

Această tabelă conține informații despre evenimentele organizate pe platformă. Fiecare înregistrare este identificată prin cheia primară id_eveniment și include numele evenimentului (nume) și data evenimentului (data_eveniment). Aceste informații sunt importante pentru planificarea și gestionarea evenimentelor.

3.14 Entitatea Mesaj

Această tabelă conține informații despre mesajele trimise între utilizatori. Fiecare înregistrare este identificată prin cheia primară id_mesaj și include referințe către utilizatorul care a trimis mesajul (id_utilizator_sender), utilizatorul care a primit mesajul (id_utilizator_receiver), conținutul mesajului (continut) și data trimiterii (data_trimitere). Aceste informații sunt cruciale pentru facilitarea comunicării private între utilizatori.

4 Descrierea relațiilor, incluzând precizarea cardinalității acestora.

4.0.1 Tabela Utilizator

Tabela Utilizator are relații de tipul **one-to-many** cu următoarele entități:

- **Postari:** Un utilizator poate crea mai multe postări (1:M(0)). (Utilizator.id_utilizator → Postari.id_utilizator)
- **Like:** Un utilizator poate da like la mai multe postări (1:M(0)). (Utilizator.id_utilizator → Like.id_utilizator)
- **Comentarii:** Un utilizator poate face mai multe comentarii (1:M(0)). (Utilizator.id_utilizator → Comentarii.id_utilizator)

- **Follow:** Un utilizator poate urmări mai mulți utilizatori și poate fi urmat de mai mulți utilizatori (1:M(0)). (`Utilizator.id.utilizator` → `Follow.id.utilizator`)
- **ParticipantEveniment:** Un utilizator poate participa la mai multe evenimente (1:M). (`Utilizator.id.utilizator` → `ParticipantEveniment.id.utilizator`)
- **MembruGrup:** Un utilizator poate fi membru în mai multe grupuri (1:M(0)). (`Utilizator.id.utilizator` → `MembruGrup.id.utilizator`)
- **Mesaj:** Un utilizator poate trimite și primi mai multe mesaje (1:M(0)). (`Utilizator.id.utilizator` → `Mesaj.id.utilizator_sender`, `Utilizator.id.utilizator` → `Mesaj.id.utilizator_receiver`)

4.0.2 Tabela Postari

Tabela `Postari` are relații de tipul `one-to-many` și `many-to-one` cu următoarele entități:

- **Utilizator:** Un utilizator poate crea mai multe postări (M(0):1). (`Postari.id.utilizator` → `Utilizator.id.utilizator`)
- **Video:** O postare poate avea un singur video (1:M(0)). (`Postari.id.postari` → `Video.id.postari`)
- **Poze:** O postare poate avea una sau mai multe poze (1:M(0)). (`Postari.id.postari` → `Poze.id.postari`)
- **Comentarii:** O postare poate avea mai multe comentarii (1:M(0)). (`Postari.id.postari` → `Comentarii.id.postari`)
- **Reactie:** O postare poate avea mai multe reacții (1:M(0)). (`Postari.id.postari` → `Reactie.id.postari`)
- **Tag:** O postare poate avea mai multe tag-uri (1:M(0)). (`Postari.id.postari` → `Tag.id.postari`)
- **Like:** O postare poate primi mai multe like-uri (1:M(0)).

4.0.3 Tabela Video

Tabela `Video` are relații de tipul `many-to-one` cu următoarele entități:

- **Postari:** O postare poate avea un singur video (1:M(0)). (`Video.id.postari` → `Postari.id.postari`)

4.0.4 Tabela Like

Tabela `Like` are relații de tipul `many-to-one` cu următoarele entități:

- **Utilizator:** Un utilizator poate da mai multe like-uri (M(0):1). (`Like.id.utilizator` → `Utilizator.id.utilizator`)
- **Postari:** O postare poate primi mai multe like-uri (M(0):1). (`Like.id.postari` → `Postari.id.postari`)

4.0.5 Tabela Comentarii

Tabela Comentarii are relații de tipul **many-to-one** cu următoarele entități:

- **Utilizator:** Un utilizator poate face mai multe comentarii (M(0):1). (`Comentarii.id.utilizator` → `Utilizator.id.utilizator`)
- **Postari:** O postare poate avea mai multe comentarii (M(0):1). (`Comentarii.id.postari` → `Postari.id.postari`)

4.0.6 Tabela Reactie

Tabela Reactie are relații de tipul **many-to-one** cu următoarele entități:

- **Postari:** O postare poate avea mai multe reacții (M(0):1). (`Reactie.id.postari` → `Postari.id.postari`)

4.0.7 Tabela Tag

Tabela Tag are relații de tipul **many-to-one** cu următoarele entități:

- **Postari:** O postare poate avea mai multe tag-uri (M(0):1). (`Tag.id.postari` → `Postari.id.postari`)

4.0.8 Tabela Follow

Tabela Follow are relații de tipul **many-to-one** cu următoarele entități:

- **Utilizator:** Un utilizator poate urmări mai mulți utilizatori și poate fi urmat de mai mulți utilizatori (M(0):1). (`Follow.id.utilizator` → `Utilizator.id.utilizator`)

4.0.9 Tabela Poze

Tabela Poze are relații de tipul **many-to-one** cu următoarele entități:

- **Postari:** O postare poate avea una sau mai multe poze (M(0):1). (`Poze.id.postari` → `Postari.id.postari`)

4.0.10 Tabela ParticipantEveniment

Tabela ParticipantEveniment are relații de tipul **many-to-one** cu următoarele entități:

- **Utilizator:** Un utilizator poate participa la mai multe evenimente (M(0):1). (`ParticipantEveniment.id.utilizator` → `Utilizator.id.utilizator`)

4.0.11 Tabela MembruGrup

Tabela MembruGrup are relații de tipul **many-to-one** cu următoarele entități:

- **Grup:** Un grup poate avea mai mulți membri (M(0):1). (`MembruGrup.id.grup` → `Grup.id.grup`)
- **Utilizator:** Un utilizator poate face parte din mai multe grupuri (M(0):1). (`MembruGrup.id.utilizator` → `Utilizator.id.utilizator`)

4.0.12 Tabela Grup

Tabela Grup are relații de tipul **one-to-many** cu următoarele entități:

- **MembruGrup:** Un grup poate avea mai mulți membri (1:M(0)). (`Grup.id_grup` → `MembruGrup.id_grup`)

4.0.13 Tabela Eveniment

Tabela Eveniment nu are relații directe specificate în descriere, dar este posibil să fie asociată cu `ParticipantEveniment` pentru a urmări participanții la evenimente.

4.0.14 Tabela Mesaj

Tabela Mesaj are relații de tipul **many-to-one** cu următoarele entități:

- **Utilizator:** Un utilizator poate trimite și primi mai multe mesaje (M(0):1). (`Mesaj.id_utilizator_sender` → `Utilizator.id_utilizator`, `Mesaj.id_utilizator_receiver` → `Utilizator.id_utilizator`)

5 Descrierea atributelor, incluzand tipul de date și eventualele constrangeri, valori implicite, valori posibile ale atributelor

5.0.1 Tabela Utilizator

- **Nume:** (Varchar2(40)):Numele utilizatorului stocat sub forma de sir de caractere de lungimea 40.
- **Email:** (Varchar2(100)):Numele emailului utilizatorului stocat sub forma de sir de caractere de lungimea 100.
- **Parola:** (Varchar2(100)):Parola utilizatorului stocata sub forma de sir de caractere de lungimea 100.
- **Data_inregistrare:** (DATE):Data la care s-a efectuat inregistrarea contului

	Name	Comment	Data Type	Not Null	Computed	Default Expression	Default On Null	Position
1	ID_UTILIZATOR		NUMBER(*)	true	false		false	1
2	NUME		VARCHAR2(40)	true	false		false	2
3	EMAIL		VARCHAR2(100)	true	false		false	3
4	PAROLA		VARCHAR2(100)	true	false		false	4
5	DATA_INREGISTRARE		DATE	true	false		false	5

Tabela Postari

- **id_postari:** (INT) Cheia primară, identificator unic pentru fiecare postare.
- **id_utilizator:** (INT) Cheie externă către tabela Utilizator, referindu-se la utilizatorul care a făcut postarea.

- **Data postarii:** (DATE) Data la care a fost făcută postarea.

	Name	Comment	Data Type	Not Null	Computed	Default Expression	Default On Null	Position
1	ID_POSTARI		NUMBER(*)	true	false		false	1
2	ID_UTILIZATOR		NUMBER(*)	false	false		false	2
3	DATA_POSTARII		DATE	true	false		false	3

Tabela Video

- **id_video:** (INT) Cheia primară, identificator unic pentru fiecare video.
- **id_postari:** (INT) Cheie externă către tabela Postari, referindu-se la postarea căreia îi aparține video-ul.
- **Titlu:** (VARCHAR(255)) Titlul video-ului, stocat sub formă de șir de caractere de lungime maximă 255.
- **Durata:** (INT) Durata video-ului, stocată ca o valoare numerică pozitivă.
- **Data upload:** (DATE) Data la care video-ul a fost încărcat.
- **URL:** (VARCHAR(255)) URL-ul video-ului, stocat sub formă de șir de caractere de lungime maximă 255.

	Name	Comment	Data Type	Not Null	Computed	Default Expression	Default On Null	Position
1	ID_VIDEO		NUMBER(*)	true	false		false	1
2	ID_POSTARI		NUMBER(*)	false	false		false	2
3	TITLU		VARCHAR2(255)	true	false		false	3
4	DURATA		NUMBER(*)	true	false		false	4
5	DATA_UPLOAD		DATE	true	false		false	5
6	URL		VARCHAR2(255)	true	false		false	6

Tabela Like

- **id_like:** (INT) Cheia primară, identificator unic pentru fiecare like.
- **id_utilizator:** (INT) Cheie externă către tabela Utilizator, referindu-se la utilizatorul care a dat like-ul.
- **id_postari:** (INT) Cheie externă către tabela Postari, referindu-se la postarea care a primit like-ul.
- **Data like:** (DATE) Data la care a fost dat like-ul.

	Name	Comment	Data Type	Not Null	Computed	Default Expression	Default On Null	Position
1	ID_LIKE		NUMBER(*)	true	false		false	1
2	ID_UTILIZATOR		NUMBER(*)	false	false		false	2
3	ID_POSTARI		NUMBER(*)	false	false		false	3
4	DATA_LIKE		DATE	true	false		false	4

Tabela Comentarii

- **id_comentarii:** (INT) Cheia primară, identificator unic pentru fiecare comentariu.
- **id_utilizator:** (INT) Cheie externă către tabela Utilizator, referindu-se la utilizatorul care a făcut comentariul.
- **id_postari:** (INT) Cheie externă către tabela Postari, referindu-se la postarea care a primit comentariul.
- **Data comentariu:** (DATE) Data la care a fost făcut comentariul.
- **Continut:** (VARCHAR(255)) Conținutul comentariului, stocat sub formă de șir de caractere de lungime maximă 255, cu lungimea minimă de 1.

	Name	Comment	Data Type	Not Null	Computed	Default Expression	Default On Null	Position
1	ID_COMENTARII		NUMBER(*)	true	false		false	1
2	ID_UTILIZATOR		NUMBER(*)	false	false		false	2
3	ID_POSTARI		NUMBER(*)	false	false		false	3
4	DATA_COMENTARIU		DATE	true	false		false	4
5	CONTINUT		VARCHAR2(255)	true	false		false	5

Tabela Reactie

- **id_reactie:** (INT) Cheia primară, identificator unic pentru fiecare reacție.
- **id_postari:** (INT) Cheie externă către tabela Postari, referindu-se la postarea care a primit reacția.
- **Tip reactie:** (VARCHAR(25)) Tipul reacției, stocat sub formă de șir de caractere de lungime maximă 25, cu lungimea minimă de 1.
- **Data reactie:** (DATE) Data la care a fost dată reacția.

	Name	Comment	Data Type	Not Null	Computed	Default Expression	Default On Null	Position
1	ID_REACTIE		NUMBER(*)	true	false		false	1
2	ID_POSTARI		NUMBER(*)	false	false		false	2
3	TIP_REACTIE		VARCHAR2(25)	true	false		false	3
4	DATA_REACTIE		DATE	true	false		false	4

Tabela Tag

- **id_tag:** (INT) Cheia primară, identificator unic pentru fiecare tag.
- **id_postari:** (INT) Cheie externă către tabela Postari, referindu-se la postarea căreia îi aparține tag-ul.
- **Nume tag:** (VARCHAR(100)) Numele tag-ului, stocat sub formă de șir de caractere de lungime maximă 100, cu lungimea minimă de 1.

	Name	Comment	Data Type	Not Null	Computed	Default Expression	Default On Null	Position
1	ID_TAG		NUMBER(*)	true	false		false	1
2	ID_POSTARI		NUMBER(*)	false	false		false	2
3	NUME_TAG		VARCHAR2(100)	true	false		false	3

Tabela Follow

- **id_follow:** (INT) Cheia primară, identificator unic pentru fiecare follow.
- **id_utilizator:** (INT) Cheie externă către tabela Utilizator, referindu-se la utilizatorul care a dat follow.
- **Data follow:** (DATE) Data la care a fost dat follow-ul.

	Name	Comment	Data Type	Not Null	Computed	Default Expression	Default On Null	Position
1	ID_FOLLOW		NUMBER(*)	true	false		false	1
2	ID_UTILIZATOR		NUMBER(*)	false	false		false	2
3	DATA_FOLLOW		DATE	true	false		false	3

Tabela Poze

- **id_poze:** (INT) Cheia primară, identificator unic pentru fiecare poză.
- **id_postari:** (INT) Cheie externă către tabela Postari, referindu-se la postarea căreia îi aparține poza.
- **Nume poza:** (VARCHAR(100)) Numele pozei, stocat sub formă de șir de caractere de lungime maximă 100, cu lungimea minimă de 1.
- **Data upload:** (DATE) Data la care poza a fost încărcată.

	Name	Comment	Data Type	Not Null	Computed	Default Expression	Default On Null	Position
1	ID_POZE		NUMBER(*)	true	false		false	1
2	ID_POSTARI		NUMBER(*)	false	false		false	2
3	NUME_POZE		VARCHAR2(100)	true	false		false	3
4	DATA_UPLOAD		DATE	true	false		false	4

Tabela Grup

- **id_grup:** (INT) Cheia primară, identificator unic pentru fiecare grup.
- **Nume:** (VARCHAR(100)) Numele grupului, stocat sub formă de șir de caractere de lungime maximă 100.

	Name	Comment	Data Type	Not Null	Computed	Default Expression	Default On Null	Position
1	ID_GRUP		NUMBER(*)	true	false		false	1
2	NUME		VARCHAR2(100)	true	false		false	2

Tabela Membru_Grup

- **id_membru_grup:** (INT) Cheia primară, identificator unic pentru fiecare membru de grup.
- **id_grup:** (INT) Cheie externă către tabela Grup, referindu-se la grupul căruia îi aparține membrul.
- **id_utilizator:** (INT) Cheie externă către tabela Utilizator, referindu-se la utilizatorul care este membru al grupului.
- **Data intrare:** (DATE) Data la care utilizatorul a intrat în grup.

	Name	Comment	Data Type	Not Null	Computed	Default Expression	Default On Null	Position
1	ID_MEMBRU_GRUP		NUMBER(*)	true	false		false	1
2	ID_GRUP		NUMBER(*)	false	false		false	2
3	ID_UTILIZATOR		NUMBER(*)	false	false		false	3
4	DATA_INTRARE		DATE	true	false		false	4

Tabela Eveniment

- **id_eveniment:** (INT) Cheia primară, identificator unic pentru fiecare eveniment.
- **Nume:** (VARCHAR(255)) Numele evenimentului, stocat sub formă de șir de caractere de lungime maximă 255.
- **Data eveniment:** (DATE) Data la care are loc evenimentul.

	Name	Comment	Data Type	Not Null	Computed	Default Expression	Default On Null	Position
1	ID_EVENTIMENT		NUMBER(*)	true	false		false	1
2	NUME		VARCHAR2(255)	true	false		false	2
3	DATA_EVENTIMENT		DATE	true	false		false	3

Tabela Participant_Eventiment

- **id_participant_eventiment:** (INT) Cheia primară, identificator unic pentru fiecare participant la eveniment.
- **id_utilizator:** (INT) Cheie externă către tabela Utilizator, referindu-se la utilizatorul care participă la eveniment.
- **id_eventiment:** (INT) Cheie externă către tabela Eveniment, referindu-se la evenimentul la care participă utilizatorul.
- **Data inregistrare:** (DATE) Data la care utilizatorul s-a înregistrat pentru eveniment.

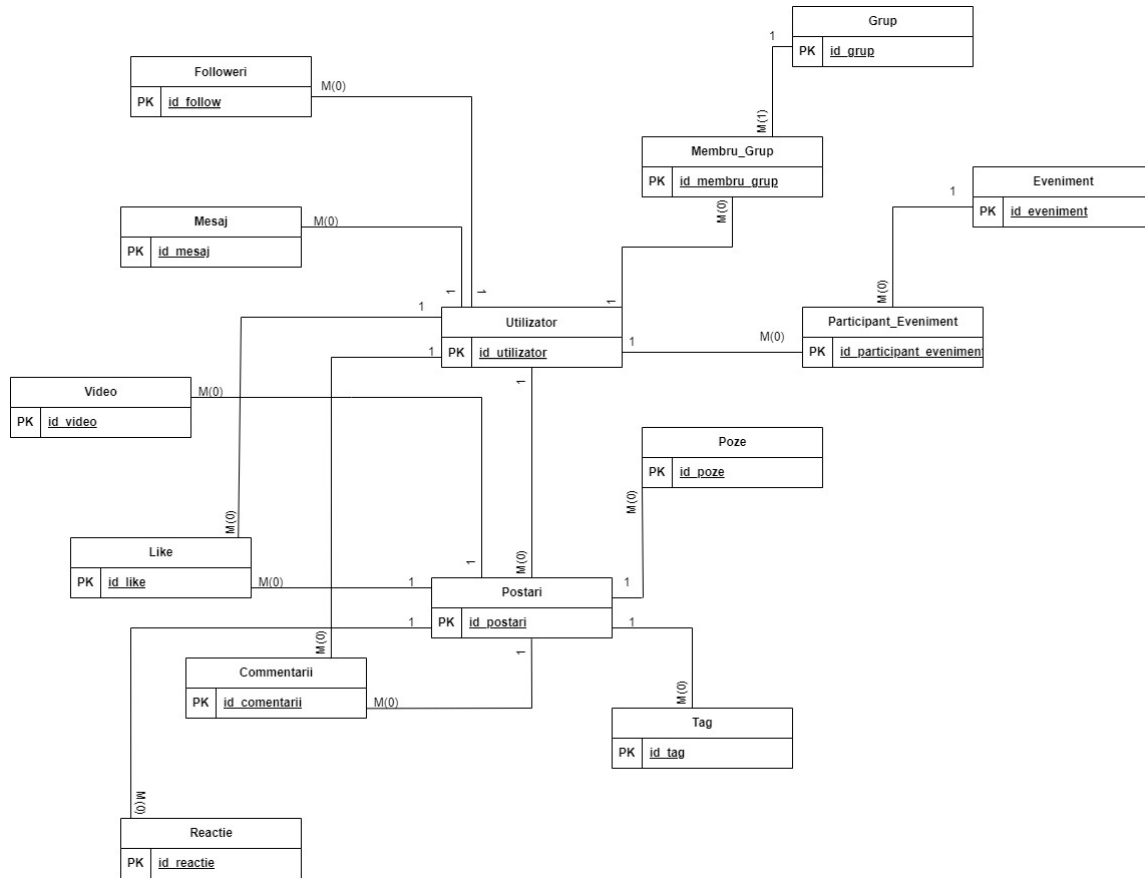
	Name	Comment	Data Type	Not Null	Computed	Default Expression	Default On Null	Position
1	ID_PARTICIPANT_EVENTIMENT		NUMBER(*)	true	false		false	1
2	ID_UTILIZATOR		NUMBER(*)	false	false		false	2
3	ID_EVENTIMENT		NUMBER(*)	false	false		false	3
4	DATA_INREGISTRARE		DATE	true	false		false	4

Tabela Mesaj

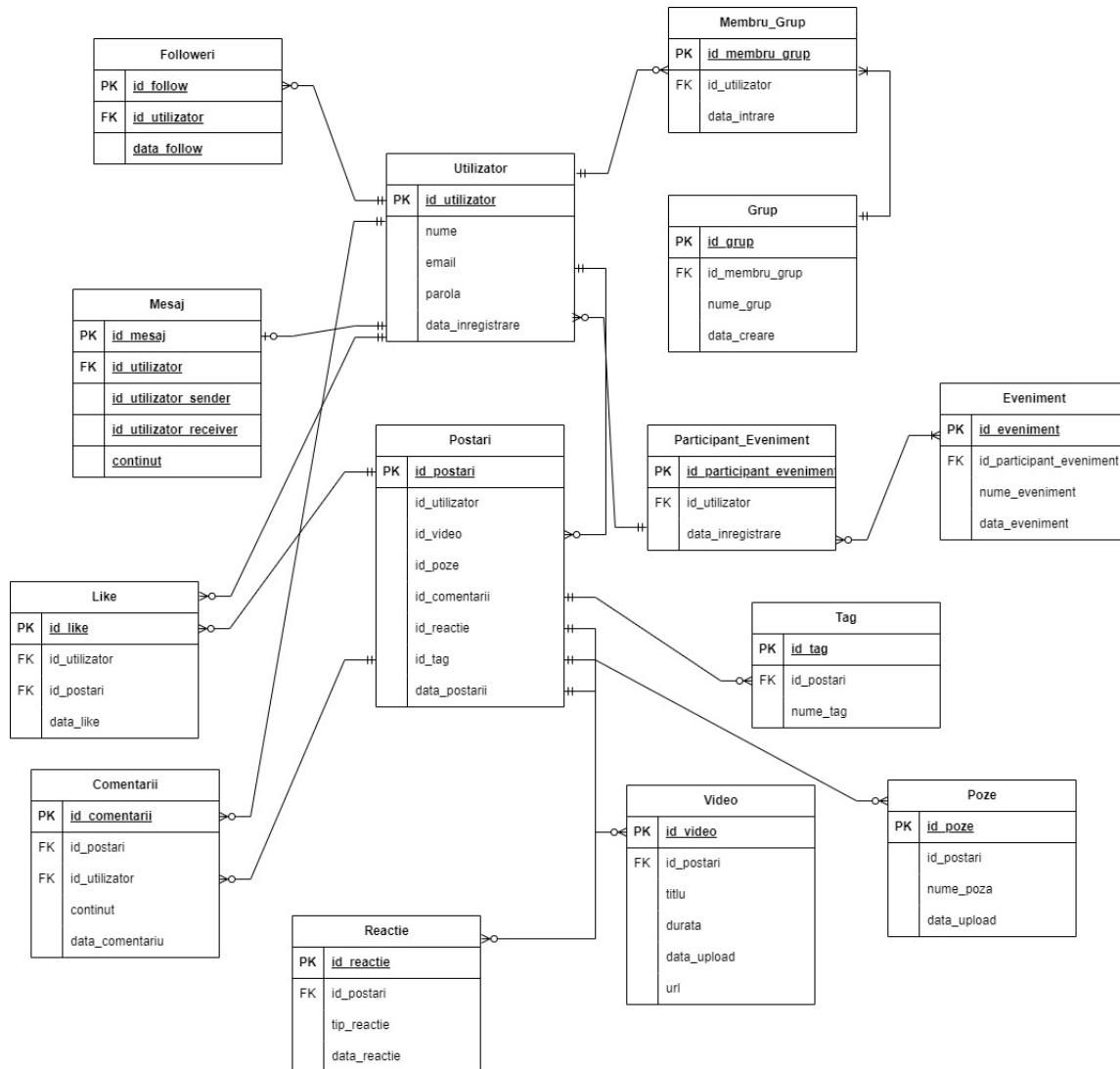
- **id_mesaj:** (INT) Cheia primară, identificator unic pentru fiecare mesaj care este de tip int.
- **id_utilizator_sender:** (INT) Cheie externă către tabela Utilizator, referindu-se la utilizatorul care a trimis mesajul.
- **id_utilizator_receiver:** (INT) Cheie externă către tabela Utilizator, referindu-se la utilizatorul care a primit mesajul.
- **Continut:** (VARCHAR(500)) Conținutul mesajului, stocat sub formă de șir de caractere de lungime maximă 500.
- **Data trimitere:** (DATE) Data la care a fost trimis mesajul.

	Name	Comment	Data Type	Not Null	Computed	Default Expression	Default On Null	Position
1	ID_MESAJ		NUMBER(*)	true	false		false	1
2	ID_UTILIZATOR_SENDER		NUMBER(*)	false	false		false	2
3	ID_UTILIZATOR_RECEIVER		NUMBER(*)	false	false		false	3
4	CONTINUT		VARCHAR2(500)	true	false		false	4
5	DATA_TRIMITERE		DATE	true	false		false	5

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



7 Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6



8 Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7

- **Utilizator**(id_utilizator, nume, email, parola, data_inregistrare)
- **Postari**(id_postari, id_utilizator, data_postarii)

- **Video**(id_video, id_postari, titlu, durata, data_upload, url)
- **LIKE**(id_like, id_utilizator, id_postari, data_like)
- **Comentarii**(comentarii, id_utilizator, id_postari, data_comentariu, continut)
- **Reactie**(id_reactie, id_postari, tip_reactie, data_reactie)
- **Tag**(id_tag, id_postari, nume_tag)
- **Follow**(id_follow, id_utilizator, data_follow)
- **Poze**(id_poze, id_postari, nume_poze, data_upload)
- **Grup**(id_grup, nume)
- **Membru_Grup**(id_membru_grup, id_grup, id_utilizator, data_intrare)
- **Eveniment**(id_eveniment, nume, data_eveniment)
- **Participant_Eventiment**(id_participant_eventiment, id_utilizator, id_eveniment, data_inregistrare)
- **Mesaj**(id_mesaj, id_utilizator_sender, id_utilizator_receiver, continut, data_trimitere)

9 Normalizarea tabelelor până la forma normală 3 (FN1-FN3)

Aceste tabele sunt normalizate până la forma normală 3 (FN1-FN3) conform următoarelor principii de normalizare:

Prima Formă Normală (FN1)

FN1 cere ca fiecare coloană a tabelului să conțină doar valori atomice (fără liste sau seturi de valori multiple) și ca fiecare rând să fie unic. În tabelele prezentate:

- Fiecare coloană conține valori atomice.
- Fiecare tabel are o cheie primară care asigură unicitatea fiecărui rând.

A Doua Formă Normală (FN2)

FN2 cere ca tabelul să fie în FN1 și, de asemenea, să nu existe dependențe parțiale ale coloanelor ne-cheie de cheie primară compusă. În tabelele prezentate:

- Tabelele care au cheie primară simplă (nu compusă) sunt automat în FN2.
- Tabelele cu cheie primară compusă (dacă ar exista) ar trebui să nu aibă atribute ne-cheie care să depindă doar parțial de cheie primară compusă.

Toate tabelele din schema dată au chei primare simple, deci nu există dependențe parțiale.

A Treia Formă Normală (FN3)

FN3 cere ca tabelul să fie în FN2 și, de asemenea, să nu existe dependențe tranzitive între coloanele ne-cheie și cheie primară. În tabelele prezentate:

- **Utilizator:** Toate coloanele depind direct de `id_utilizator`. Nu există dependențe tranzitive.
- **Postari:** Toate coloanele depind direct de `id_postari`. `id_utilizator` este o referință către tabelul `Utilizator`, dar nu există dependențe tranzitive.
- **Video:** Toate coloanele depind direct de `id_video`. `id_postari` este o referință către tabelul `Postari`, dar nu există dependențe tranzitive.
- **LIKE:** Toate coloanele depind direct de `id_like`. `id_utilizator` și `id_postari` sunt referințe către tabelele `Utilizator` și `Postari`, dar nu există dependențe tranzitive.
- **Comentarii:** Toate coloanele depind direct de `id_comentarii`. `id_utilizator` și `id_postari` sunt referințe către tabelele `Utilizator` și `Postari`, dar nu există dependențe tranzitive.
- **Reactie:** Toate coloanele depind direct de `id_reactie`. `id_postari` este o referință către tabelul `Postari`, dar nu există dependențe tranzitive.
- **Tag:** Toate coloanele depind direct de `id_tag`. `id_postari` este o referință către tabelul `Postari`, dar nu există dependențe tranzitive.
- **Follow:** Toate coloanele depind direct de `id_follow`. `id_utilizator` este o referință către tabelul `Utilizator`, dar nu există dependențe tranzitive.
- **Poze:** Toate coloanele depind direct de `id_poze`. `id_postari` este o referință către tabelul `Postari`, dar nu există dependențe tranzitive.
- **Grup:** Toate coloanele depind direct de `id_grup`. Nu există dependențe tranzitive.
- **Membru_Grup:** Toate coloanele depind direct de `id_membru_grup`. `id_grup` și `id_utilizator` sunt referințe către tabelele `Grup` și `Utilizator`, dar nu există dependențe tranzitive.
- **Eveniment:** Toate coloanele depind direct de `id_eveniment`. Nu există dependențe tranzitive.
- **Participant_Eveniment:** Toate coloanele depind direct de `id_participant_eveniment`. `id_utilizator` și `id_eveniment` sunt referințe către tabelele `Utilizator` și `Eveniment`, dar nu există dependențe tranzitive.
- **Mesaj:** Toate coloanele depind direct de `id_mesaj`. `id_utilizator_sender` și `id_utilizator_receiver` sunt referințe către tabelul `Utilizator`, dar nu există dependențe tranzitive.

Prin urmare, toate tabelele din schema prezentată sunt normalizate până la forma normală 3, asigurându-se astfel că datele sunt structurate într-un mod care elimină redundanța și dependențele tranzitive.

10 Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).

```
1 CREATE SEQUENCE seq_utilizator START WITH 1 INCREMENT BY 1;
2 CREATE SEQUENCE seq_postari START WITH 1 INCREMENT BY 1;
3 CREATE SEQUENCE seq_video START WITH 1 INCREMENT BY 1;
4 CREATE SEQUENCE seq_like START WITH 1 INCREMENT BY 1;
5 CREATE SEQUENCE seq_comentarii START WITH 1 INCREMENT BY 1;
6 CREATE SEQUENCE seq_reactie START WITH 1 INCREMENT BY 1;
7 CREATE SEQUENCE seq_tag START WITH 1 INCREMENT BY 1;
8 CREATE SEQUENCE seq_follow START WITH 1 INCREMENT BY 1;
9 CREATE SEQUENCE seq_poze START WITH 1 INCREMENT BY 1;
10 CREATE SEQUENCE seq_grup START WITH 1 INCREMENT BY 1;
11 CREATE SEQUENCE seq_membru_grup START WITH 1 INCREMENT BY 1;
12 CREATE SEQUENCE seq_eveniment START WITH 1 INCREMENT BY 1;
13 CREATE SEQUENCE seq_participant_eveniment START WITH 1 INCREMENT BY 1;
14 CREATE SEQUENCE seq_mesaj START WITH 1 INCREMENT BY 1;
15
16 CREATE OR REPLACE TRIGGER trg_utilizator
17 BEFORE INSERT ON Utilizator
18 FOR EACH ROW
19 BEGIN
20     SELECT seq_utilizator.NEXTVAL INTO :NEW.id_utilizator FROM dual;
21 END;
22 /
23
24 CREATE OR REPLACE TRIGGER trg_postari
25 BEFORE INSERT ON Postari
26 FOR EACH ROW
27 BEGIN
28     SELECT seq_postari.NEXTVAL INTO :NEW.id_postari FROM dual;
29 END;
30 /
31
32 CREATE OR REPLACE TRIGGER trg_video
33 BEFORE INSERT ON Video
34 FOR EACH ROW
35 BEGIN
36     SELECT seq_video.NEXTVAL INTO :NEW.id_video FROM dual;
37 END;
38 /
39
40 CREATE OR REPLACE TRIGGER trg_like
41 BEFORE INSERT ON "LIKE"
42 FOR EACH ROW
43 BEGIN
44     SELECT seq_like.NEXTVAL INTO :NEW.id_like FROM dual;
45 END;
46 /
```

```

47
48 CREATE OR REPLACE TRIGGER trg_comentarii
49 BEFORE INSERT ON Comentarii
50 FOR EACH ROW
51 BEGIN
52     SELECT seq_comentarii.NEXTVAL INTO :NEW.id_comentarii FROM dual;
53 END;
54 /
55
56 CREATE OR REPLACE TRIGGER trg_reactie
57 BEFORE INSERT ON Reactie
58 FOR EACH ROW
59 BEGIN
60     SELECT seq_reactie.NEXTVAL INTO :NEW.id_reactie FROM dual;
61 END;
62 /
63
64 CREATE OR REPLACE TRIGGER trg_tag
65 BEFORE INSERT ON Tag
66 FOR EACH ROW
67 BEGIN
68     SELECT seq_tag.NEXTVAL INTO :NEW.id_tag FROM dual;
69 END;
70 /
71
72 CREATE OR REPLACE TRIGGER trg_follow
73 BEFORE INSERT ON Follow
74 FOR EACH ROW
75 BEGIN
76     SELECT seq_follow.NEXTVAL INTO :NEW.id_follow FROM dual;
77 END;
78 /
79
80 CREATE OR REPLACE TRIGGER trg_poze
81 BEFORE INSERT ON Poze
82 FOR EACH ROW
83 BEGIN
84     SELECT seq_poze.NEXTVAL INTO :NEW.id_poze FROM dual;
85 END;
86 /
87
88 CREATE OR REPLACE TRIGGER trg_grup
89 BEFORE INSERT ON Grup
90 FOR EACH ROW
91 BEGIN
92     SELECT seq_grup.NEXTVAL INTO :NEW.id_grup FROM dual;
93 END;
94 /
95
96 CREATE OR REPLACE TRIGGER trg_membru_grup

```



```

197 BEFORE INSERT ON Membru_Grup
198 FOR EACH ROW
199 BEGIN
200     SELECT seq_membru_grup.NEXTVAL INTO :NEW.id_membru_grup FROM dual;
201 END;
202 /
203
204 CREATE OR REPLACE TRIGGER trg_eveniment
205 BEFORE INSERT ON Eveniment
206 FOR EACH ROW
207 BEGIN
208     SELECT seq_eveniment.NEXTVAL INTO :NEW.id_eveniment FROM dual;
209 END;
210 /
211
212 CREATE OR REPLACE TRIGGER trg_participant_eveniment
213 BEFORE INSERT ON Participant_Eveniment
214 FOR EACH ROW
215 BEGIN
216     SELECT seq_participant_eveniment.NEXTVAL INTO :NEW.
217         id_participant_eveniment FROM dual;
218 END;
219 /
220
221 CREATE OR REPLACE TRIGGER trg_mesaj
222 BEFORE INSERT ON Mesaj
223 FOR EACH ROW
224 BEGIN
225     SELECT seq_mesaj.NEXTVAL INTO :NEW.id_mesaj FROM dual;
226 END;

```

Aceste triggere asigură faptul ca fiecare înregistrare are un ID unic, incrementat automat, fără a fi nevoie ca utilizatorul să furnizeze manual un ID la momentul inserării.

```
UTILIZATOR> CREATE SEQUENCE seq_postari START WITH 1 INCREMENT BY 1
[2024-05-24 12:59:35] completed in 2 ms
UTILIZATOR> CREATE SEQUENCE seq_video START WITH 1 INCREMENT BY 1
[2024-05-24 12:59:35] completed in 3 ms
UTILIZATOR> CREATE SEQUENCE seq_like START WITH 1 INCREMENT BY 1
[2024-05-24 12:59:35] completed in 2 ms
UTILIZATOR> CREATE SEQUENCE seq_comentarii START WITH 1 INCREMENT BY 1
[2024-05-24 12:59:35] completed in 3 ms
UTILIZATOR> CREATE SEQUENCE seq_reactie START WITH 1 INCREMENT BY 1
[2024-05-24 12:59:35] completed in 2 ms
UTILIZATOR> CREATE SEQUENCE seq_tag START WITH 1 INCREMENT BY 1
[2024-05-24 12:59:35] completed in 2 ms
UTILIZATOR> CREATE SEQUENCE seq_follow START WITH 1 INCREMENT BY 1
[2024-05-24 12:59:35] completed in 2 ms
UTILIZATOR> CREATE SEQUENCE seq_poze START WITH 1 INCREMENT BY 1
[2024-05-24 12:59:35] completed in 2 ms
UTILIZATOR> CREATE SEQUENCE seq_grup START WITH 1 INCREMENT BY 1
[2024-05-24 12:59:35] completed in 2 ms
UTILIZATOR> CREATE SEQUENCE seq_membru_grup START WITH 1 INCREMENT BY 1
[2024-05-24 12:59:35] completed in 2 ms
UTILIZATOR> CREATE SEQUENCE seq_eveniment START WITH 1 INCREMENT BY 1
[2024-05-24 12:59:35] completed in 2 ms
UTILIZATOR> CREATE SEQUENCE seq_participant_eveniment START WITH 1 INCREMENT BY 1
[2024-05-24 12:59:35] completed in 2 ms
UTILIZATOR> CREATE SEQUENCE seq_mesaj START WITH 1 INCREMENT BY 1
[2024-05-24 12:59:35] completed in 3 ms
```

```

UTILIZATOR> CREATE OR REPLACE TRIGGER trg_utilizator
    BEFORE INSERT ON Utilizator
    FOR EACH ROW
    BEGIN
        SELECT seq_utilizator.NEXTVAL INTO :NEW.id_utilizator FROM dual;
    END;
[2024-05-24 12:59:35] completed in 10 ms
UTILIZATOR> CREATE OR REPLACE TRIGGER trg_postari
    BEFORE INSERT ON Postari
    FOR EACH ROW
    BEGIN
        SELECT seq_postari.NEXTVAL INTO :NEW.id_postari FROM dual;
    END;
[2024-05-24 12:59:35] completed in 7 ms
UTILIZATOR> CREATE OR REPLACE TRIGGER trg_video
    BEFORE INSERT ON Video
    FOR EACH ROW
    BEGIN
        SELECT seq_video.NEXTVAL INTO :NEW.id_video FROM dual;
    END;
[2024-05-24 12:59:35] completed in 8 ms
UTILIZATOR> CREATE OR REPLACE TRIGGER trg_like
    BEFORE INSERT ON "LIKE"
    FOR EACH ROW
    BEGIN
        SELECT seq_like.NEXTVAL INTO :NEW.id_like FROM dual;
    END;
[2024-05-24 12:59:35] completed in 8 ms
UTILIZATOR> CREATE OR REPLACE TRIGGER trg_comentarii
    BEFORE INSERT ON Comentarii
    FOR EACH ROW
    BEGIN
        SELECT seq_comentarii.NEXTVAL INTO :NEW.id_comentarii FROM dual;
    END;

```

```

UTILIZATOR> CREATE OR REPLACE TRIGGER trg_reactie
    BEFORE INSERT ON Reactie
    FOR EACH ROW
    BEGIN
        SELECT seq_reactie.NEXTVAL INTO :NEW.id_reactie FROM dual;
    END;
[2024-05-24 12:59:35] completed in 9 ms
UTILIZATOR> CREATE OR REPLACE TRIGGER trg_tag
    BEFORE INSERT ON Tag
    FOR EACH ROW
    BEGIN
        SELECT seq_tag.NEXTVAL INTO :NEW.id_tag FROM dual;
    END;
[2024-05-24 12:59:35] completed in 7 ms
UTILIZATOR> CREATE OR REPLACE TRIGGER trg_follow
    BEFORE INSERT ON Follow
    FOR EACH ROW
    BEGIN
        SELECT seq_follow.NEXTVAL INTO :NEW.id_follow FROM dual;
    END;
[2024-05-24 12:59:35] completed in 7 ms
UTILIZATOR> CREATE OR REPLACE TRIGGER trg_poze
    BEFORE INSERT ON Poze
    FOR EACH ROW
    BEGIN
        SELECT seq_poze.NEXTVAL INTO :NEW.id_poze FROM dual;
    END;
[2024-05-24 12:59:35] completed in 10 ms
UTILIZATOR> CREATE OR REPLACE TRIGGER trg_grup
    BEFORE INSERT ON Grup
    FOR EACH ROW
    BEGIN
        SELECT seq_grup.NEXTVAL INTO :NEW.id_grup FROM dual;
    END;
[2024-05-24 12:59:35] completed in 9 ms
UTILIZATOR> CREATE OR REPLACE TRIGGER trg_membru_grup
    BEFORE INSERT ON Membru_Grup
    FOR EACH ROW
    BEGIN
        SELECT seq_membru_grup.NEXTVAL INTO :NEW.id_membru_grup FROM dual;
    END;

```

```

UTILIZATOR> CREATE OR REPLACE TRIGGER trg_eveniment
              BEFORE INSERT ON Eveniment
              FOR EACH ROW
              BEGIN
                  SELECT seq_eveniment.NEXTVAL INTO :NEW.id_eveniment FROM dual;
              END;
[2024-05-24 12:59:35] completed in 9 ms
UTILIZATOR> CREATE OR REPLACE TRIGGER trg_participant_eveniment
              BEFORE INSERT ON Participant_Eveniment
              FOR EACH ROW
              BEGIN
                  SELECT seq_participant_eveniment.NEXTVAL INTO :NEW.id_participant_eveniment FROM dual;
              END;
[2024-05-24 12:59:35] completed in 7 ms
UTILIZATOR> CREATE OR REPLACE TRIGGER trg_mesaj
              BEFORE INSERT ON Mesaj
              FOR EACH ROW
              BEGIN
                  SELECT seq_mesaj.NEXTVAL INTO :NEW.id_mesaj FROM dual;
              END;
[2024-05-24 12:59:35] completed in 8 ms

```

11 Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea

11.1 Pentru Tabela Utilizator

Creare Tabel

```

1 CREATE TABLE Utilizator (
2     id_utilizator INT PRIMARY KEY,
3     nume VARCHAR(40) NOT NULL,
4     email VARCHAR(100) NOT NULL,
5     parola VARCHAR(100) NOT NULL CHECK (LENGTH(parola) >= 1),
6     data_inregistrare DATE NOT NULL
7 );

1 INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
2 VALUES ('Ion Popescu', 'ion.popescu@gmail.com', 'parola123', DATE '
    2024-05-05');
3
4 INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
5 VALUES ('Maria Ionescu', 'maria.ionescu@gmail.com', 'qwreg12I2K341FK',
    DATE '2022-12-12');
6
7 INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
8 VALUES ('Stancu Rares', 'rares.stancu@gmail.com', 'rares?!!', DATE '
    2023-10-01');

```

```

9
10 INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
11 VALUES ('Ana Popa', 'ana.popa@gmail.com', 'parola', DATE '2023-04-01');
12
13 INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
14 VALUES ('Madalin Ioana', 'ioana.madalin@gmail.com', 'madalin123', DATE '
    2024-09-09');
15
16 INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
17 VALUES ('Elena Petrescu', 'elena.petrescu@gmail.com', '2IBB1C12sx', DATE
    '2021-04-02');
18
19 INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
20 VALUES ('Mihai Dumitru', 'mihai.dumitru@gmail.com', '16232651242', DATE
    '2019-11-11');
21
22 INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
23 VALUES ('Cristina Radu', 'cristina.radu@gmail.com', 'nb3i0o2bf2vy', DATE
    '2024-05-02');
24
25 INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
26 VALUES ('Florin Stoica', 'florin.stoica@gmail.com', '3b4fgf43p12b', DATE
    '2023-09-01');
27
28 INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
29 VALUES ('Ioana Pavel', 'ioana.pavel@gmail.com', 'VJ23GTvg33o080HVC',
    DATE '2020-01-03');
30
31 SELECT * FROM UTILIZATOR ORDER BY ID_UTILIZATOR;

```

ID_UTILIZATOR	NUME	EMAIL	PAROLA	DATA_INREGISTRARE
1	Ion Popescu	ion.popescu@gmail.com	parola123	2024-05-05
2	Maria Ionescu	maria.ionescu@gmail.com	qwreg12I2K341FK	2022-12-12
3	Stancu Rares	rares.stancu@gmail.com	rares?!!	2023-10-01
4	Ana Popa	ana.popa@gmail.com	parola	2023-04-01
5	Madalin Ioana	ioana.madalin@gmail.com	madalin123	2024-09-09
6	Elena Petrescu	elena.petrescu@gmail.com	2IBB1C12sx	2021-04-02
7	Mihai Dumitru	mihai.dumitru@gmail.com	16232651242	2019-11-11
8	Cristina Radu	cristina.radu@gmail.com	nb3i0o2bf2vy	2024-05-02
9	Florin Stoica	florin.stoica@gmail.com	3b4fgf43p12b	2023-09-01
10	Ioana Pavel	ioana.pavel@gmail.com	VJ23GTvg33o080HVC	2020-01-03

```
1 ✓ INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
2 VALUES ('Ion Popescu', 'ion.popescu@gmail.com', 'parola123', DATE '2024-05-05');
3
4 ✓ INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
5 VALUES ('Maria Ionescu', 'maria.ionescu@gmail.com', 'qwreg12I2K341FK', DATE '2022-12-12');
6
7 ✓ INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
8 VALUES ('Stancu Rares', 'rares.stancu@gmail.com', 'rares?!!', DATE '2023-10-01');
9
10 ✓ INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
11 VALUES ('Ana Popa', 'ana.popa@gmail.com', 'parola', DATE '2023-04-01');
12
13 ✓ INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
14 VALUES ('Mădalin Ioana', 'ioana.mădalin@gmail.com', 'mădalin123', DATE '2024-09-09');
15
16 ✓ INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
17 VALUES ('Elena Petrescu', 'elena.petrescu@gmail.com', '2IBB1C12sx', DATE '2021-04-02');
18
19 ✓ INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
20 VALUES ('Mihai Dumitru', 'mihai.dumitru@gmail.com', '16232651242', DATE '2019-11-11');
21
22 ✓ INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
23 VALUES ('Cristina Radu', 'cristina.radu@gmail.com', 'nb3i0o2bf2vy', DATE '2024-05-02');
24
25 ✓ INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
26 VALUES ('Florin Stoica', 'florin.stoica@gmail.com', '3b4fgf43p12b', DATE '2023-09-01');
27
28 ✓ INSERT INTO Utilizator (nume, email, parola, data_inregistrare)
29 VALUES ('Ioana Pavel', 'ioana.pavel@gmail.com', 'VJ236Tvg33o080HVC', DATE '2020-01-03');
30
31 ✓ SELECT * FROM UTILIZATOR ORDER BY ID_UTILIZATOR;
```

11.2 Pentru Tabela Postari

Creare Tabel

```
1 CREATE TABLE Postari (
2     id_postari INT PRIMARY KEY,
3     id_utilizator INT,
4     data_postarii DATE NOT NULL,
5     FOREIGN KEY (id_utilizator) REFERENCES Utilizator(id_utilizator)
6 );
```

```
1 INSERT INTO Postari (id_utilizator, data_postarii)
```

```
2 VALUES (1, DATE '2024-10-10');
3
4 INSERT INTO Postari (id_utilizator, data_postarii)
5 VALUES (2, DATE '2024-01-15');
6
7 INSERT INTO Postari (id_utilizator, data_postarii)
8 VALUES (3, DATE '2023-12-12');
9
10 INSERT INTO Postari (id_utilizator, data_postarii)
11 VALUES (4, DATE '2024-05-25');
12
13 INSERT INTO Postari (id_utilizator, data_postarii)
14 VALUES (5, DATE '2023-12-01');
15
16 INSERT INTO Postari (id_utilizator, data_postarii)
17 VALUES (6, DATE '2022-03-13');
18
19 INSERT INTO Postari (id_utilizator, data_postarii)
20 VALUES (7, DATE '2024-02-02');
21
22 INSERT INTO Postari (id_utilizator, data_postarii)
23 VALUES (8, DATE '2023-09-15');
24
25 INSERT INTO Postari (id_utilizator, data_postarii)
26 VALUES (9, DATE '2023-07-20');
27
28 INSERT INTO Postari (id_utilizator, data_postarii)
29 VALUES (10, DATE '2023-11-26');
30
31 SELECT * FROM POSTARI ORDER BY ID_POSTARI;
```


	ID_POSTARI	ID_UTILIZATOR	DATA_POSTARII
1	1	1	2024-10-10
2	2	2	2024-01-15
3	3	3	2023-12-12
4	4	4	2024-05-25
5	5	5	2023-12-01
6	6	6	2022-03-13
7	7	7	2024-02-02
8	8	8	2023-09-15
9	9	9	2023-07-20
10	10	10	2023-11-26

11.3 Pentru Tabela Video

Creare Tabel

```
1 CREATE TABLE Video (  
2     id_video INT PRIMARY KEY,  
3     id_postari INT,  
4     titlu VARCHAR(255) NOT NULL,  
5     durata INT NOT NULL CHECK (durata > 0),  
6     data_upload DATE NOT NULL,  
7     url VARCHAR(255) NOT NULL,  
8     FOREIGN KEY (id_postari) REFERENCES Postari(id_postari)  
9 );
```

```
1 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)  
2 VALUES (1, 'Munte', 30, DATE '2024-10-10', 'http://mysocial.com/video1')  
3 ;  
4 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)  
5 VALUES (2, 'Tutorial programare', 300, DATE '2024-01-15', 'http://  
6     mysocial.com/video2');  
7 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)  
8 VALUES (3, 'Concert live', 180, DATE '2023-12-12', 'http://mysocial.com/  
9     video3');  
10 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)  
11 VALUES (4, 'Re et de buc t rie', 240, DATE '2024-05-25', 'http://  
12     mysocial.com/video4');  
13 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)  
14 VALUES (5, 'Documentar natura', 360, DATE '2023-12-01', 'http://mysocial  
15     .com/video5');  
16 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)  
17 VALUES (6, 'Exerci ii fitness', 150, DATE '2022-03-13', 'http://  
18     mysocial.com/video6');  
19 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)  
20 VALUES (7, 'Tutorial', 200, DATE '2024-02-02', 'http://mysocial.com/  
21     video7');  
22 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)  
23 VALUES (8, 'Prezentare proiect', 90, DATE '2023-09-15', 'http://mysocial  
24     .com/video8');  
25 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)  
26 VALUES (9, 'Interviu', 180, DATE '2023-07-20', 'http://mysocial.com/  
27     video9');  
28 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)
```

```

29 VALUES (10, 'Vlog', 100, DATE '2023-11-26', 'http://mysocial.com/video10
    ');
30
31 SELECT * FROM VIDEO ORDER BY ID_VIDEO;

```

ID_VIDEO	ID_POSTARI	TITLU	DURATA	DATA_UPLOAD	URL
1	1	1 Munte	30	2024-10-10	http://mysocial.com/video1
2	2	2 Tutorial programare	300	2024-01-15	http://mysocial.com/video2
3	3	3 Concert live	180	2023-12-12	http://mysocial.com/video3
4	4	4 Rețetă de bucătărie	240	2024-05-25	http://mysocial.com/video4
5	5	5 Documentar natura	360	2023-12-01	http://mysocial.com/video5
6	6	6 Exerciții fitness	150	2022-03-13	http://mysocial.com/video6
7	7	7 Tutorial	200	2024-02-02	http://mysocial.com/video7
8	8	8 Prezentare proiect	90	2023-09-15	http://mysocial.com/video8
9	9	9 Interviu	180	2023-07-20	http://mysocial.com/video9
10	10	10 Vlog	100	2023-11-26	http://mysocial.com/video10

```
1 ✓ INSERT INTO Video (id_postari, titlu, durata, data_upload, url)
2 VALUES (1, 'Munte', 30, DATE '2024-10-10', 'http://mysocial.com/video1');
3
4 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)
5 VALUES (2, 'Tutorial programare', 300, DATE '2024-01-15', 'http://mysocial.com/video2');
6
7 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)
8 VALUES (3, 'Concert live', 180, DATE '2023-12-12', 'http://mysocial.com/video3');
9
10 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)
11 VALUES (4, 'Retetă de bucătărie', 240, DATE '2024-05-25', 'http://mysocial.com/video4');
12
13 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)
14 VALUES (5, 'Documentar natura', 360, DATE '2023-12-01', 'http://mysocial.com/video5');
15
16 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)
17 VALUES (6, 'Exercitii fitness', 150, DATE '2022-03-13', 'http://mysocial.com/video6');
18
19 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)
20 VALUES (7, 'Tutorial', 200, DATE '2024-02-02', 'http://mysocial.com/video7');
21
22 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)
23 VALUES (8, 'Prezentare proiect', 90, DATE '2023-09-15', 'http://mysocial.com/video8');
24
25 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)
26 VALUES (9, 'Interviu', 180, DATE '2023-07-20', 'http://mysocial.com/video9');
27
28 INSERT INTO Video (id_postari, titlu, durata, data_upload, url)
29 VALUES (10, 'Vlog', 100, DATE '2023-11-26', 'http://mysocial.com/video10');
30
31 SELECT * FROM VIDEO ORDER BY ID_VIDEO;
```

11.4 Pentru Tabela Like

Creare Tabel

```
1 CREATE TABLE "LIKE" (
2     id_like INT PRIMARY KEY,
3     id_utilizator INT,
4     id_postari INT,
5     data_like DATE NOT NULL,
6     FOREIGN KEY (id_utilizator) REFERENCES Utilizator(id_utilizator),
7     FOREIGN KEY (id_postari) REFERENCES Postari(id_postari)
8 );
```

```
1 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
2 VALUES (1, 1, DATE '2024-10-11');
3
4 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
5 VALUES (1, 2, DATE '2024-01-14');
6
7 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
8 VALUES (3, 2, DATE '2024-01-15');
9
10 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
11 VALUES (4, 4, DATE '2024-05-26');
12
13 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
14 VALUES (5, 5, DATE '2023-12-02');
15
16 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
17 VALUES (6, 6, DATE '2022-03-14');
18
19 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
20 VALUES (7, 7, DATE '2024-02-03');
21
22 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
23 VALUES (8, 8, DATE '2023-09-16');
24
25 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
26 VALUES (9, 9, DATE '2023-07-20');
27
28 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
29 VALUES (10, 10, DATE '2023-11-26');
30
31 SELECT * FROM "LIKE" ORDER BY ID_LIKE;
```

	ID_LIKE	ID_UTILIZATOR	ID_POSTARI	DATA_LIKE
1	1	1	1	2024-10-11
2	2	1	2	2024-01-14
3	3	3	2	2024-01-15
4	4	4	4	2024-05-26
5	5	5	5	2023-12-02
6	6	6	6	2022-03-14
7	7	7	7	2024-02-03
8	8	8	8	2023-09-16
9	9	9	9	2023-07-20
10	10	10	10	2023-11-26

```
1  INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
2  VALUES (1, 1, DATE '2024-10-11');
3
4  INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
5  VALUES (1, 2, DATE '2024-01-14');
6
7  INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
8  VALUES (3, 2, DATE '2024-01-15');
9
10 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
11 VALUES (4, 4, DATE '2024-05-26');
12
13 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
14 VALUES (5, 5, DATE '2023-12-02');
15
16 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
17 VALUES (6, 6, DATE '2022-03-14');
18
19 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
20 VALUES (7, 7, DATE '2024-02-03');
21
22 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
23 VALUES (8, 8, DATE '2023-09-16');
24
25 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
26 VALUES (9, 9, DATE '2023-07-20');
27
28 INSERT INTO "LIKE" (id_utilizator, id_postari, data_like)
29 VALUES (10, 10, DATE '2023-11-26');
30
31 ✓ SELECT * FROM "LIKE" ORDER BY ID_LIKE;
```

11.5 Pentru Tabela Comentarii

Creare Tabel

```
1 CREATE TABLE Comentarii (  
2     id_comentarii INT PRIMARY KEY,  
3     id_utilizator INT,  
4     id_postari INT,  
5     data_comentariu DATE NOT NULL,  
6     continut VARCHAR(255) NOT NULL CHECK (LENGTH(continut) >= 1),  
7     FOREIGN KEY (id_utilizator) REFERENCES Utilizator(id_utilizator),  
8     FOREIGN KEY (id_postari) REFERENCES Postari(id_postari)  
9 );
```

```
1 INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu,  
2     continut)  
3 VALUES (1, 1, DATE '2024-10-11', 'Frumos video!');  
4  
5 INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu,  
6     continut)  
7 VALUES (2, 2, DATE '2024-10-13', 'Foarte util tutorialul!');  
8  
9 INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu,  
10    continut)  
11 VALUES (3, 3, DATE '2024-01-15', 'Excelent Concert!!');  
12  
13 INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu,  
14    continut)  
15 VALUES (4, 4, DATE '2024-05-26', 'Foarte Buna reteta');  
16  
17 INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu,  
18    continut)  
19 VALUES (5, 5, DATE '2022-03-14', 'Foarte interesant documentar-ul.');20  
21 INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu,  
22    continut)  
23 VALUES (6, 6, DATE '2024-02-03', 'Exercitiile au fost foarte eficiente');  
24  
25 INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu,  
26    continut)  
27 VALUES (7, 7, DATE '2023-09-16', 'Foarte Util');28  
29 INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu,  
30    continut)  
31 VALUES (8, 8, DATE '2024-02-17', 'Foarte bine facut!');32  
33 INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu,  
34    continut)  
35 VALUES (9, 9, DATE '2023-07-20', 'Foarte interesant interviul.');36
```



```

28 INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu,
    continut)
29 VALUES (10, 10, DATE '2023-11-26', ' mi place vlog-ul t u!');
30
31 SELECT * FROM COMENTARII ORDER BY ID_COMENTARII;

```

	ID_COMENTARII	ID_UTILIZATOR	ID_POSTARI	DATA_COMENTARIU	CONTINUT
1	1	1	1	2023-01-12	Frumos video!
2	2	2	2	2023-01-17	Foarte util tutorialul!
3	3	3	3	2023-01-22	Excelent Concert!!
4	4	4	4	2023-01-27	Foarte Buna reteta
5	5	5	5	2023-02-03	Foarte interesant documentar-ul.
6	6	6	6	2023-02-07	Exercitiile au fost foarte eficiente
7	7	7	7	2023-02-12	Foarte Util
8	8	8	8	2023-02-17	Foarte bine facut!
9	9	9	9	2023-02-22	Foarte interesant interviul.
10	10	10	10	2023-02-27	Îmi place vlog-ul tău!

```
1 ✓ INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu, continut)
2   VALUES (1, 1, DATE '2023-01-12', 'Frumos video!');
3
4 ✓ INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu, continut)
5   VALUES (2, 2, DATE '2023-01-17', 'Foarte util tutorialul!');
6
7 ✓ INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu, continut)
8   VALUES (3, 3, DATE '2023-01-22', 'Excelent Concert!!');
9
10 ✓ INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu, continut)
11   VALUES (4, 4, DATE '2023-01-27', 'Foarte Buna reteta');
12
13 ✓ INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu, continut)
14   VALUES (5, 5, DATE '2023-02-03', 'Foarte interesant documentar-ul. ');
15
16 ✓ INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu, continut)
17   VALUES (6, 6, DATE '2023-02-07', 'Exercitiile au fost foarte eficiente');
18
19 ✓ INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu, continut)
20   VALUES (7, 7, DATE '2023-02-12', 'Foarte Util');
21
22 ✓ INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu, continut)
23   VALUES (8, 8, DATE '2023-02-17', 'Foarte bine facut!');
24
25 ✓ INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu, continut)
26   VALUES (9, 9, DATE '2023-02-22', 'Foarte interesant interviul. ');
27
28 ✓ INSERT INTO Comentarii (id_utilizator, id_postari, data_comentariu, continut)
29   VALUES (10, 10, DATE '2023-02-27', 'Imi place vlog-ul tau!');
30
31 ✓ SELECT * FROM COMENTARII ORDER BY ID_COMENTARII;
```

11.6 Pentru Tabela Reactie

Creare Tabel

```
1 CREATE TABLE Reactie (
2   id_reactie INT PRIMARY KEY,
3   id_postari INT,
4   tip_reactie VARCHAR(25) NOT NULL CHECK (LENGTH(tip_reactie) >= 1),
5   data_reactie DATE NOT NULL,
```

```

6      FOREIGN KEY (id_postari) REFERENCES Postari(id_postari)
7  );

```

```

1  INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
2  VALUES (1, 'Love', DATE '2024-10-11');
3
4  INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
5  VALUES (2, 'Wow', DATE '2024-10-13');
6
7  INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
8  VALUES (3, 'Like', DATE '2024-01-15');
9
10 INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
11 VALUES (4, 'Haha', DATE '2024-05-26');
12
13 INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
14 VALUES (5, 'Sad', DATE '2022-03-14');
15
16 INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
17 VALUES (6, 'Angry', DATE '2024-02-03');
18
19 INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
20 VALUES (7, 'Haha', DATE '2023-09-16');
21
22 INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
23 VALUES (8, 'Wow', DATE '2024-02-17');
24
25 INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
26 VALUES (9, 'Like', DATE '2023-11-26');
27
28 INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
29 VALUES (10, 'Haha', DATE '2023-02-28');
30
31 SELECT * FROM REACTIE ORDER BY ID_REACTIE;

```

	ID_REACTIE	ID_POSTARI	TIP_REACTIE	DATA_REACTIE
1	1	1	Love	2024-10-11
2	2	2	Wow	2024-10-13
3	3	3	Like	2024-01-15
4	4	4	Haha	2024-05-26
5	5	5	Sad	2022-03-14
6	6	6	Angry	2024-02-03
7	7	7	Haha	2023-09-16
8	8	8	Wow	2024-02-17
9	9	9	Like	2023-11-26
10	10	10	Haha	2023-02-28

```
1 ✓ INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
2 VALUES (1, 'Love', DATE '2024-10-11');
3
4 ✓ INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
5 VALUES (2, 'Wow', DATE '2024-10-13');
6
7 ✓ INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
8 VALUES (3, 'Like', DATE '2024-01-15');
9
10 ✓ INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
11 VALUES (4, 'Haha', DATE '2024-05-26');
12
13 ✓ INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
14 VALUES (5, 'Sad', DATE '2022-03-14');
15
16 ✓ INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
17 VALUES (6, 'Angry', DATE '2024-02-03');
18
19 ✓ INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
20 VALUES (7, 'Haha', DATE '2023-09-16');
21
22 ✓ INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
23 VALUES (8, 'Wow', DATE '2024-02-17');
24
25 ✓ INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
26 VALUES (9, 'Like', DATE '2023-11-26');
27
28 ✓ INSERT INTO Reactie (id_postari, tip_reactie, data_reactie)
29 VALUES (10, 'Haha', DATE '2023-02-28');
30
31 ✓ SELECT * FROM REACTIE ORDER BY ID_REACTIE;
```



11.7 Pentru Tabela Tag




Creare Tabel


```
1 CREATE TABLE Tag (  
2     id_tag INT PRIMARY KEY,  
3     id_postari INT,  
4     nume_tag VARCHAR(100) NOT NULL CHECK (LENGTH(nume_tag) >= 1),  
5     FOREIGN KEY (id_postari) REFERENCES Postari(id_postari)  
6 );
```




```
1 INSERT INTO Tag (id_postari, nume_tag)  
2 VALUES (1, 'vacan ');  
3  
4 INSERT INTO Tag (id_postari, nume_tag)  
5 VALUES (2, 'programare');  
6  
7 INSERT INTO Tag (id_postari, nume_tag)  
8 VALUES (3, 'muzic ');  
9  
10 INSERT INTO Tag (id_postari, nume_tag)  
11 VALUES (4, 'buc t rie');  
12  
13 INSERT INTO Tag (id_postari, nume_tag)  
14 VALUES (5, 'natur ');  
15  
16 INSERT INTO Tag (id_postari, nume_tag)  
17 VALUES (6, 'fitness');  
18  
19 INSERT INTO Tag (id_postari, nume_tag)  
20 VALUES (7, 'c l torii');  
21  
22 INSERT INTO Tag (id_postari, nume_tag)  
23 VALUES (8, 'produs');  
24  
25 INSERT INTO Tag (id_postari, nume_tag)  
26 VALUES (9, 'interviu');  
27  
28 INSERT INTO Tag (id_postari, nume_tag)  
29 VALUES (10, 'vlog');  
30  
31 SELECT * FROM TAG ORDER BY ID_TAG;
```

	ID_TAG	ID_POSTARI	NUME_TAG
1	1	1	vacanță
2	2	2	programare
3	3	3	muzică
4	4	4	bucătărie
5	5	5	natură
6	6	6	fitness
7	7	7	călătorii
8	8	8	produs
9	9	9	interviu
10	10	10	vlog









Tx: Manual 

PI

1 


2

3

4 


5

6

7 


8

9

10 


11

12

13 


14

15

16 


17

18

19 


20

21

22 

23

24

25 

26

27

```
INSERT INTO Tag (id_postari, nume_tag)
VALUES (1, 'vacantă');

INSERT INTO Tag (id_postari, nume_tag)
VALUES (2, 'programare');

INSERT INTO Tag (id_postari, nume_tag)
VALUES (3, 'muzică');

INSERT INTO Tag (id_postari, nume_tag)
VALUES (4, 'bucătărie');

INSERT INTO Tag (id_postari, nume_tag)
VALUES (5, 'natură');

INSERT INTO Tag (id_postari, nume_tag)
VALUES (6, 'fitness');

INSERT INTO Tag (id_postari, nume_tag)
VALUES (7, 'călătorii');

INSERT INTO Tag (id_postari, nume_tag)
VALUES (8, 'produs');

INSERT INTO Tag (id_postari, nume_tag)
VALUES (9, 'interviu');
```


11.8 Pentru Tabela Follow

Creare Tabel

```
1 CREATE TABLE Follow (  
2     id_follow INT PRIMARY KEY,  
3     id_utilizator INT,  
4     data_follow DATE NOT NULL,  
5     FOREIGN KEY (id_utilizator) REFERENCES Utilizator(id_utilizator)  
6 );
```

```
1 INSERT INTO Follow (id_utilizator, data_follow)  
2 VALUES (1, DATE '2024-10-11');  
3  
4 INSERT INTO Follow (id_utilizator, data_follow)  
5 VALUES (2, DATE '2024-10-13');  
6  
7 INSERT INTO Follow (id_utilizator, data_follow)  
8 VALUES (3, DATE '2024-01-15');  
9  
10 INSERT INTO Follow (id_utilizator, data_follow)  
11 VALUES (4, DATE '2024-05-26');  
12  
13 INSERT INTO Follow (id_utilizator, data_follow)  
14 VALUES (5, DATE '2022-03-14');  
15  
16 INSERT INTO Follow (id_utilizator, data_follow)  
17 VALUES (6, DATE '2024-02-03');  
18  
19 INSERT INTO Follow (id_utilizator, data_follow)  
20 VALUES (7, DATE '2023-09-16');  
21  
22 INSERT INTO Follow (id_utilizator, data_follow)  
23 VALUES (8, DATE '2024-02-17');  
24  
25 INSERT INTO Follow (id_utilizator, data_follow)  
26 VALUES (9, DATE '2023-07-20');  
27  
28 INSERT INTO Follow (id_utilizator, data_follow)  
29 VALUES (10, DATE '2023-11-26');  
30  
31 SELECT * FROM FOLLOW ORDER BY ID_FOLLOW;
```

	ID_FOLLOW	ID_UTILIZATOR	DATA_FOLLOW
1	1	1	2024-10-11
2	2	2	2024-10-13
3	3	3	2024-01-15
4	4	4	2024-05-26
5	5	5	2022-03-14
6	6	6	2024-02-03
7	7	7	2023-09-16
8	8	8	2024-02-17
9	9	9	2023-07-20
10	10	10	2023-11-26

11.9 Pentru Tabela Poze

Creare Tabel

```
1 CREATE TABLE Poze (  
2     id_poze INT PRIMARY KEY,  
3     id_postari INT,  
4     nume_poze VARCHAR(100) NOT NULL CHECK (LENGTH(nume_poze) >= 1),  
5     data_upload DATE NOT NULL,  
6     FOREIGN KEY (id_postari) REFERENCES Postari(id_postari)  
7 );
```

```
1 INSERT INTO Poze (id_postari, nume_poze, data_upload)  
2 VALUES (1, 'vacanta.jpg', DATE '2024-10-11');  
3  
4 INSERT INTO Poze (id_postari, nume_poze, data_upload)  
5 VALUES (2, 'tutorial.jpg', DATE '2024-10-13');  
6  
7 INSERT INTO Poze (id_postari, nume_poze, data_upload)  
8 VALUES (3, 'concert.jpg', DATE '2024-01-15');  
9  
10 INSERT INTO Poze (id_postari, nume_poze, data_upload)  
11 VALUES (4, 'reteta.jpg', DATE '2024-05-26');  
12  
13 INSERT INTO Poze (id_postari, nume_poze, data_upload)  
14 VALUES (5, 'documentar.jpg', DATE '2022-03-14');  
15  
16 INSERT INTO Poze (id_postari, nume_poze, data_upload)  
17 VALUES (6, 'exercitii.jpg', DATE '2024-02-03');  
18  
19 INSERT INTO Poze (id_postari, nume_poze, data_upload)  
20 VALUES (7, 'ghid.jpg', DATE '2023-09-16');  
21  
22 INSERT INTO Poze (id_postari, nume_poze, data_upload)  
23 VALUES (8, 'prezentare.jpg', DATE '2024-02-17');  
24  
25 INSERT INTO Poze (id_postari, nume_poze, data_upload)  
26 VALUES (9, 'interviu.jpg', DATE '2023-07-20');  
27  
28 INSERT INTO Poze (id_postari, nume_poze, data_upload)  
29 VALUES (10, 'vlog.jpg', DATE '2023-11-26');  
30  
31 SELECT * FROM POZE ORDER BY ID_POZE;
```

	ID_POZE	ID_POSTARI	NUME_POZE	DATA_UPLOAD
1	1	1	vacanta.jpg	2024-10-11
2	2	2	tutorial.jpg	2024-10-13
3	3	3	concert.jpg	2024-01-15
4	4	4	reteta.jpg	2024-05-26
5	5	5	documentar.jpg	2022-03-14
6	6	6	exercitii.jpg	2024-02-03
7	7	7	ghid.jpg	2023-09-16
8	8	8	prezentare.jpg	2024-02-17
9	9	9	interviu.jpg	2023-07-20
10	10	10	vlog.jpg	2023-11-26

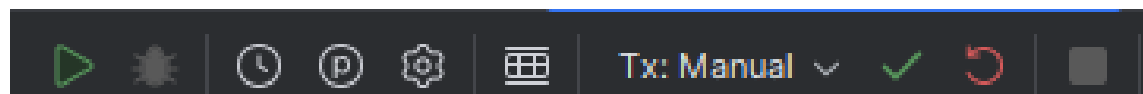
11.10 Pentru Tabela Grup

Creare Tabel

```
1 CREATE TABLE Grup (  
2     id_grup INT PRIMARY KEY,  
3     nume VARCHAR(100) NOT NULL  
4 );
```

```
1 INSERT INTO Grup (nume)  
2 VALUES ('Grup de prieteni');  
3  
4 INSERT INTO Grup (nume)  
5 VALUES ('Grup de programatori');  
6  
7 INSERT INTO Grup (nume)  
8 VALUES ('Grup de muzicieni');  
9  
10 INSERT INTO Grup (nume)  
11 VALUES ('Grup de buc tari');  
12  
13 INSERT INTO Grup (nume)  
14 VALUES ('Grup de naturali ti');  
15  
16 INSERT INTO Grup (nume)  
17 VALUES ('Grup de sportivi');  
18  
19 INSERT INTO Grup (nume)  
20 VALUES ('Grup de c l tori');  
21  
22 INSERT INTO Grup (nume)  
23 VALUES ('Grup de marketeri');  
24  
25 INSERT INTO Grup (nume)  
26 VALUES ('Grup de jurnali ti');  
27  
28 INSERT INTO Grup (nume)  
29 VALUES ('Grup de vlogeri');  
30  
31 SELECT * FROM GRUP ORDER BY ID_GRUP;
```

	ID_GRP		NUME	
1		1	Grup de prieteni	
2		2	Grup de programatori	
3		3	Grup de muzicieni	
4		4	Grup de bucătari	
5		5	Grup de naturaliști	
6		6	Grup de sportivi	
7		7	Grup de călători	
8		8	Grup de marketeri	
9		9	Grup de jurnaliști	
10		10	Grup de vlogeri	



```
1  ✓  INSERT INTO Grup (nume)
2      VALUES ('Grup de prieteni');
3
4      INSERT INTO Grup (nume)
5      VALUES ('Grup de programatori');
6
7      INSERT INTO Grup (nume)
8      VALUES ('Grup de muzicieni');
9
10     INSERT INTO Grup (nume)
11     VALUES ('Grup de bucătari');
12
13     INSERT INTO Grup (nume)
14     VALUES ('Grup de naturalisti');
15
16     INSERT INTO Grup (nume)
17     VALUES ('Grup de sportivi');
18
19     INSERT INTO Grup (nume)
20     VALUES ('Grup de călători');
21
22     INSERT INTO Grup (nume)
23     VALUES ('Grup de marketeri');
24
25     INSERT INTO Grup (nume)
```

11.11 Pentru Tabela Membru_{Grup}

Creare Tabel

```
1 CREATE TABLE Membru_Grup (  
2     id_membru_grup INT PRIMARY KEY,  
3     id_grup INT,  
4     id_utilizator INT,  
5     data_intrare DATE NOT NULL,  
6     FOREIGN KEY (id_grup) REFERENCES Grup(id_grup),  
7     FOREIGN KEY (id_utilizator) REFERENCES Utilizator(id_utilizator)  
8 );
```

```
1 INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)  
2 VALUES (1, 1, DATE '2024-10-11');  
3  
4 INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)  
5 VALUES (2, 2, DATE '2024-10-13');  
6  
7 INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)  
8 VALUES (3, 3, DATE '2024-01-15');  
9  
10 INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)  
11 VALUES (4, 4, DATE '2024-05-26');  
12  
13 INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)  
14 VALUES (5, 5, DATE '2022-03-14');  
15  
16 INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)  
17 VALUES (6, 6, DATE '2024-02-03');  
18  
19 INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)  
20 VALUES (7, 7, DATE '2023-09-16');  
21  
22 INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)  
23 VALUES (8, 8, DATE '2024-02-17');  
24  
25 INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)  
26 VALUES (9, 9, DATE '2023-07-20');  
27  
28 INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)  
29 VALUES (10, 10, DATE '2023-11-26');  
30  
31 SELECT * FROM MEMBRU_GRP ORDER BY ID_MEMBRU_GRP;
```

	WHERE		ORDER BY		
	ID_MEMBRU_GRP	ID_GRP	ID_UTILIZATOR	DATA_INTRARE	
1	1	1	1	2024-10-11	
2	2	2	2	2024-10-13	
3	3	3	3	2024-01-15	
4	4	4	4	2024-05-26	
5	5	5	5	2022-03-14	
6	6	6	6	2024-02-03	
7	7	7	7	2023-09-16	
8	8	8	8	2024-02-17	
9	9	9	9	2023-07-20	
10	10	10	10	2023-11-26	

```
1 ✓ ▾ INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)
2     VALUES (1, 1, DATE '2024-10-11');
3
4 ▾ INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)
5     VALUES (2, 2, DATE '2024-10-13');
6
7 ▾ INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)
8     VALUES (3, 3, DATE '2024-01-15');
9
10 ▾ INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)
11     VALUES (4, 4, DATE '2024-05-26');
12
13 ▾ INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)
14     VALUES (5, 5, DATE '2022-03-14');
15
16 ▾ INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)
17     VALUES (6, 6, DATE '2024-02-03');
18
19 ▾ INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)
20     VALUES (7, 7, DATE '2023-09-16');
21
22 ▾ INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)
23     VALUES (8, 8, DATE '2024-02-17');
24     💡
25 ▾ INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)
26     VALUES (9, 9, DATE '2023-07-20');
27
28 ▾ INSERT INTO Membru_Grup (id_grup, id_utilizator, data_intrare)
29     VALUES (10, 10, DATE '2023-11-26');
30
31 SELECT * FROM MEMBRU_GRP ORDER BY ID_MEMBRU_GRP;
```

11.12 Pentru Tabela Eveniment

Creare Tabel

```
1 CREATE TABLE Eveniment (  
2     id_eveniment INT PRIMARY KEY,  
3     nume VARCHAR(255) NOT NULL,  
4     data_eveniment DATE NOT NULL  
5 );
```

```
1 INSERT INTO Eveniment (nume, data_eveniment)  
2 VALUES ('Festival Muzical', DATE '2024-12-11');  
3  
4 INSERT INTO Eveniment (nume, data_eveniment)  
5 VALUES ('Hackathon', DATE '2024-12-01');  
6  
7 INSERT INTO Eveniment (nume, data_eveniment)  
8 VALUES ('Expozi ie de Art ', DATE '2024-03-15');  
9  
10 INSERT INTO Eveniment (nume, data_eveniment)  
11 VALUES ('Maraton', DATE '2024-09-26');  
12  
13 INSERT INTO Eveniment (nume, data_eveniment)  
14 VALUES ('Festival de Film', DATE '2022-06-11');  
15  
16 INSERT INTO Eveniment (nume, data_eveniment)  
17 VALUES ('Competi ie de gaming', DATE '2024-12-04');  
18  
19 INSERT INTO Eveniment (nume, data_eveniment)  
20 VALUES ('Curs de G tit', DATE '2023-10-16');  
21  
22 INSERT INTO Eveniment (nume, data_eveniment)  
23 VALUES ('Noaptea Muzeelor', DATE '2024-05-17');  
24  
25 INSERT INTO Eveniment (nume, data_eveniment)  
26 VALUES ('Concurs Fitness', DATE '2023-07-21');  
27  
28 INSERT INTO Eveniment (nume, data_eveniment)  
29 VALUES ('Campionat de ah ', DATE '2023-12-26');  
30  
31 SELECT * FROM EVENIMENT ORDER BY ID_EVENIMENT;
```

10 rows			
ID_EVENTIMENT	NUME	DATA_EVENTIMENT	
1	Festival Muzical	2024-12-11	
2	Hackathon	2024-12-01	
3	Expoziție de Artă	2024-03-15	
4	Maraton	2024-09-26	
5	Festival de Film	2022-06-11	
6	Competiție de gaming	2024-12-04	
7	Curs de Gătit	2023-10-16	
8	Noaptea Muzeelor	2024-05-17	
9	Concurs Fitness	2023-07-21	
10	Campionat de Șah	2023-12-26	

Tx: Manual ✓ ↺

Playground ▾

```
1 ✓ ▾ INSERT INTO Eveniment (nume, data_eveniment)  
2     VALUES ('Festival Muzical', DATE '2024-12-11');  
3 ✓   COMMIT;  
4 ✓ ▾ INSERT INTO Eveniment (nume, data_eveniment)  
5     VALUES ('Hackathon', DATE '2024-12-01');  
6 ✓   COMMIT;  
7 ✓ ▾ INSERT INTO Eveniment (nume, data_eveniment)  
8     VALUES ('Expozitie de Artă', DATE '2024-03-15');  
9 ✓   COMMIT;  
10 ✓ ▾ INSERT INTO Eveniment (nume, data_eveniment)  
11    VALUES ('Maraton', DATE '2024-09-26');  
12 ✓   COMMIT;  
13 ✓ ▾ INSERT INTO Eveniment (nume, data_eveniment)  
14    VALUES ('Festival de Film', DATE '2022-06-11');  
15 ✓   COMMIT;  
16 ✓ ▾ INSERT INTO Eveniment (nume, data_eveniment)  
17    VALUES ('Competitie de gaming', DATE '2024-12-04');  
18 ✓   COMMIT;  
19 ✓ ▾ INSERT INTO Eveniment (nume, data_eveniment)  
20    VALUES ('Curs de Gătire', DATE '2023-10-16');  
21 ✓   COMMIT;  
22 ✓ ▾ INSERT INTO Eveniment (nume, data_eveniment)  
23    VALUES ('Noaptea Muzeelor', DATE '2024-05-17');  
24 ✓   COMMIT;  
25 ✓ ▾ INSERT INTO Eveniment (nume, data_eveniment)  
26    VALUES ('Concurs Fitness', DATE '2023-07-21');  
27 ✓   COMMIT;  
28 ✓ ▾ INSERT INTO Eveniment (nume, data_eveniment)  
29    VALUES ('Campionat de Șah', DATE '2023-12-26');  
30 ✓   COMMIT;  
31  
32 ✓ SELECT * FROM EVENIMENT ORDER BY ID_EVENTIMENT;
```

11.13 Pentru Tabela Participant_{Eveniment}

Creare Tabel

```
1 CREATE TABLE Participant_Eventiment (  
2     id_participant_eventiment INT PRIMARY KEY,  
3     id_utilizator INT,  
4     id_eventiment INT,  
5     data_inregistrare DATE NOT NULL,  
6     FOREIGN KEY (id_utilizator) REFERENCES Utilizator(id_utilizator),  
7     FOREIGN KEY (id_eventiment) REFERENCES Eveniment(id_eventiment)  
8 );
```

```
1 INSERT INTO Participant_Eventiment (id_utilizator, id_eventiment,  
2     data_inregistrare)  
3 VALUES (1, 1, DATE '2024-10-11');  
4  
5 INSERT INTO Participant_Eventiment (id_utilizator, id_eventiment,  
6     data_inregistrare)  
7 VALUES (2, 2, DATE '2024-10-13');  
8  
9 INSERT INTO Participant_Eventiment (id_utilizator, id_eventiment,  
10    data_inregistrare)  
11 VALUES (3, 3, DATE '2024-01-15');  
12  
13 INSERT INTO Participant_Eventiment (id_utilizator, id_eventiment,  
14    data_inregistrare)  
15 VALUES (4, 4, DATE '2024-05-26');  
16  
17 INSERT INTO Participant_Eventiment (id_utilizator, id_eventiment,  
18    data_inregistrare)  
19 VALUES (5, 5, DATE '2022-03-14');  
20  
21 INSERT INTO Participant_Eventiment (id_utilizator, id_eventiment,  
22    data_inregistrare)  
23 VALUES (6, 6, DATE '2024-02-03');  
24  
25 INSERT INTO Participant_Eventiment (id_utilizator, id_eventiment,  
26    data_inregistrare)  
27 VALUES (7, 7, DATE '2023-09-16');  
28  
29 INSERT INTO Participant_Eventiment (id_utilizator, id_eventiment,  
30    data_inregistrare)  
31 VALUES (8, 8, DATE '2024-02-17');  
32  
33 INSERT INTO Participant_Eventiment (id_utilizator, id_eventiment,  
34    data_inregistrare)  
35 VALUES (9, 9, DATE '2023-07-20');  
36  
37 INSERT INTO Participant_Eventiment (id_utilizator, id_eventiment,  
38    data_inregistrare)
```



```

29 VALUES (10, 10, DATE '2023-11-26');
30
31 SELECT * FROM PARTICIPANT_EVENTIMENT ORDER BY ID_PARTICIPANT_EVENTIMENT;

```

	ID_PARTICIPANT_EVENTIMENT	ID_UTILIZATOR	ID_EVENTIMENT	DATA_INREGISTRARE
1	1	1	1	2024-10-11
2	2	2	2	2024-10-13
3	3	3	3	2024-01-15
4	4	4	4	2024-05-26
5	5	5	5	2022-03-14
6	6	6	6	2024-02-03
7	7	7	7	2023-09-16
8	8	8	8	2024-02-17
9	9	9	9	2023-07-20
10	10	10	10	2023-11-26

```
1 ✓ INSERT INTO Participant_Eventiment (id_utilizator, id_eveniment, data_inregistrare)
2 VALUES (1, 1, DATE '2024-10-11');
3 ✓ COMMIT;
4 ✓ INSERT INTO Participant_Eventiment (id_utilizator, id_eveniment, data_inregistrare)
5 VALUES (2, 2, DATE '2024-10-13');
6 ✓ COMMIT;
7 ✓ INSERT INTO Participant_Eventiment (id_utilizator, id_eveniment, data_inregistrare)
8 VALUES (3, 3, DATE '2024-01-15');
9 ✓ COMMIT;
10 ✓ INSERT INTO Participant_Eventiment (id_utilizator, id_eveniment, data_inregistrare)
11 VALUES (4, 4, DATE '2024-05-26');
12 ✓ COMMIT;
13 ✓ INSERT INTO Participant_Eventiment (id_utilizator, id_eveniment, data_inregistrare)
14 VALUES (5, 5, DATE '2022-03-14');
15 ✓ COMMIT;
16 ✓ INSERT INTO Participant_Eventiment (id_utilizator, id_eveniment, data_inregistrare)
17 VALUES (6, 6, DATE '2024-02-03');
18 ✓ COMMIT;
19 ✓ INSERT INTO Participant_Eventiment (id_utilizator, id_eveniment, data_inregistrare)
20 VALUES (7, 7, DATE '2023-09-16');
21 ✓ COMMIT;
22 ✓ INSERT INTO Participant_Eventiment (id_utilizator, id_eveniment, data_inregistrare)
23 VALUES (8, 8, DATE '2024-02-17');
24 ✓ COMMIT;
25 ✓ INSERT INTO Participant_Eventiment (id_utilizator, id_eveniment, data_inregistrare)
26 VALUES (9, 9, DATE '2023-07-20');
27 ✓ COMMIT;
28 ✓ INSERT INTO Participant_Eventiment (id_utilizator, id_eveniment, data_inregistrare)
29 VALUES (10, 10, DATE '2023-11-26');
30 ✓ COMMIT;
31
32 ✓ SELECT * FROM PARTICIPANT_EVENTIMENT ORDER BY ID_PARTICIPANT_EVENTIMENT;
```

11.14 Pentru Tabela Mesaj

Creare Tabel

```
1 CREATE TABLE Mesaj (
2     id_mesaj INT PRIMARY KEY,
3     id_utilizator_sender INT,
4     id_utilizator_receiver INT,
5     continut VARCHAR(500) NOT NULL,
6     data_trimitere DATE NOT NULL,
```

```

7      FOREIGN KEY (id_utilizator_sender) REFERENCES Utilizator(
          id_utilizator),
8      FOREIGN KEY (id_utilizator_receiver) REFERENCES Utilizator(
          id_utilizator)
9  );

```

```

1  INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver,
    continut, data_trimitere)
2  VALUES (1, 2, 'Salut! Ce mai faci?', DATE '2024-10-12 ');
3
4  INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver,
    continut, data_trimitere)
5  VALUES (2, 3, 'Vrei s mergem la film?', DATE '2024-10-14');
6
7  INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver,
    continut, data_trimitere)
8  VALUES (3, 4, 'Am o ntrebare despre proiect.', DATE '2024-01-16');
9
10 INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver,
    continut, data_trimitere)
11 VALUES (4, 5, 'Ne vedem m ine?', DATE '2024-05-26');
12
13 INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver,
    continut, data_trimitere)
14 VALUES (5, 6, 'Cum a fost vacan a?', DATE '2022-03-15');
15
16 INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver,
    continut, data_trimitere)
17 VALUES (6, 7, 'Ce ai mai f cut?', DATE '2024-02-04');
18
19 INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver,
    continut, data_trimitere)
20 VALUES (7, 8, 'Ai terminat raportul?', DATE '2023-09-17');
21
22 INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver,
    continut, data_trimitere)
23 VALUES (8, 9, 'Mergem la concert?', DATE '2024-02-18');
24
25 INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver,
    continut, data_trimitere)
26 VALUES (9, 10, 'Ce p rere ai despre film?', DATE '2023-07-21');
27
28 INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver,
    continut, data_trimitere)
29 VALUES (10, 1, 'Vrei s ie im n weekend?', DATE '2023-11-27');
30
31 SELECT * FROM MESAJ ORDER BY ID_MESAJ;

```

ID_MESAJ	ID_UTILIZATOR_SENDER	ID_UTILIZATOR_RECEIVER	CONTINUT	DATA_TRIMITERE
1	1	1	2 Salut! Ce mai faci?	2024-10-12
2	2	2	3 Vrei să mergem la film?	2024-10-14
3	3	3	4 Am o întrebare despre proiect.	2024-01-16
4	4	4	5 Ne vedem mâine?	2024-05-26
5	5	5	6 Cum a fost vacanța?	2022-03-15
6	6	6	7 Ce ai mai făcut?	2024-02-04
7	7	7	8 Ai terminat raportul?	2023-09-17
8	8	8	9 Mergem la concert?	2024-02-18
9	9	9	10 Ce părere ai despre film?	2023-07-21
10	10	10	1 Vrei să ieșim în weekend?	2023-11-27

```

1 ✓ INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver, continut, data_trimitere)
2   VALUES (1, 2, 'Salut! Ce mai faci?', DATE '2024-10-12 ');
3 ✓ COMMIT;
4 ✓ INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver, continut, data_trimitere)
5   VALUES (2, 3, 'Vrei să mergem la film?', DATE '2024-10-14');
6 ✓ COMMIT;
7 ✓ INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver, continut, data_trimitere)
8   VALUES (3, 4, 'Am o întrebare despre proiect.', DATE '2024-01-16');
9 ✓ COMMIT;
10 ✓ INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver, continut, data_trimitere)
11   VALUES (4, 5, 'Ne vedem mâine?', DATE '2024-05-26');
12 ✓ COMMIT;
13 ✓ INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver, continut, data_trimitere)
14   VALUES (5, 6, 'Cum a fost vacanța?', DATE '2022-03-15');
15 ✓ COMMIT;
16 ✓ INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver, continut, data_trimitere)
17   VALUES (6, 7, 'Ce ai mai făcut?', DATE '2024-02-04');
18 ✓ COMMIT;
19 ✓ INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver, continut, data_trimitere)
20   VALUES (7, 8, 'Ai terminat raportul?', DATE '2023-09-17');
21 ✓ COMMIT;
22 ✓ INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver, continut, data_trimitere)
23   VALUES (8, 9, 'Mergem la concert?', DATE '2024-02-18');
24 ✓ COMMIT;
25 ✓ INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver, continut, data_trimitere)
26   VALUES (9, 10, 'Ce părere ai despre film?', DATE '2023-07-21');
27 ✓ COMMIT;
28 ✓ INSERT INTO Mesaj (id_utilizator_sender, id_utilizator_receiver, continut, data_trimitere)
29   VALUES (10, 1, 'Vrei să ieșim în weekend?', DATE '2023-11-27');
30 ✓ COMMIT;
31 ✓ SELECT * FROM MESAJ ORDER BY ID_MESAJ;

```

12 Formulați în limbaj natural și implementați 5 cereri SQL complexe

1. Selectează numele utilizatorilor care s-au înregistrat între anii 2020 și 2021, care au o adresă de email care conține litera 'a', și au cel puțin o postare de tip video sau un like dat unei postări. Din aceștia, selectează pe cei care sunt membri ai unui grup și care au trimis cel puțin un mesaj, apoi calculează cati followeri au acestia. În final, afișează numele și utilizatorilor, numarul de followeri si numarul de postari.

```
1  SELECT
2      U.numa ,
3      U.email ,
4      COUNT(DISTINCT F1.id_utilizator) AS followeri ,
5      COUNT(DISTINCT P.id_postari) AS numar_postari
6  FROM Utilizator U
7  JOIN
8      "LIKE" L ON U.id_utilizator = L.id_utilizator
9  JOIN
10     Postari P ON L.id_postari = P.id_postari
11  JOIN
12     Video V ON P.id_postari = V.id_postari
13  JOIN
14     Membru_Grup MG ON U.id_utilizator = MG.id_utilizator
15  JOIN
16     Mesaj M ON U.id_utilizator = M.id_utilizator_sender
17  JOIN
18     Follow F1 ON U.id_utilizator = F1.id_utilizator
19  WHERE
20     U.data_inregistrare BETWEEN DATE '2020-01-01' AND DATE '2021-12-31'
21  AND
22     U.email LIKE '%a%'
23  GROUP BY
24     U.numa , U.email
25  HAVING COUNT(DISTINCT M.id_mesaj) >= 1 AND COUNT(DISTINCT MG.
        ID_MEMBRU_GRP) >= 1 AND (COUNT(DISTINCT V.ID_VIDEO) >= 1 OR (COUNT(
        DISTINCT L.ID_LIKE) >= 1));
```

	NUME	EMAIL	NUMAR_FOLLOWERI	NUMAR_POSTARI
1	Ioana Pavel	ioana.pavel@gmail.com	1	1
2	Elena Petrescu	elena.petrescu@gmail.com	1	1

The screenshot shows a SQL IDE with a query editor on the left and a results pane on the right. The query is as follows:

```

1 SELECT
2     U.ume,
3     U.email,
4     COUNT(DISTINCT F1.id_utilizator) AS followeri,
5     COUNT(DISTINCT P.id_postari) AS numar_postari
6 FROM Utilizator U
7 JOIN
8     "LIKE" L ON U.id_utilizator = L.id_utilizator
9 JOIN
10    Postari P ON L.id_postari = P.id_postari
11 JOIN
12    Video V ON P.id_postari = V.id_postari
13 JOIN
14    Membru_Grup MG ON U.id_utilizator = MG.id_utilizator
15 JOIN
16    Mesaj M ON U.id_utilizator = M.id_utilizator_sender
17 JOIN
18    Follow F1 ON U.id_utilizator = F1.id_utilizator
19 WHERE
20    U.data_inregistrare BETWEEN DATE '2020-01-01' AND DATE '2021-12-31'
21 AND
22    U.email LIKE '%a%'
23 GROUP BY
24    U.ume, U.email
25 HAVING COUNT(DISTINCT M.id_mesaj) >= 1 AND COUNT(DISTINCT MG.ID_MEMBRU_GRUP)>=1 AND (COUNT(DISTINCT V.ID_VIDEO)>=1 OR (COUNT(DISTINCT L.ID_LIKE)>=1));

```

The results pane shows the following data:

NUME	EMAIL	FOLLOWERI	NUMAR_POSTARI
Elena Petrescu	elena.petrescu@gmail.com	1	1
Ioana Pavel	ioana.pavel@gmail.com	1	1

2. Obțineti Numele utilizatorilor, titlul videoclipurilor, durata acestora dar si numarul de likeuri al utilizatorilor care au postat videoclipuri cu o durată mai mare de 100 minute și care au primit cel puțin 2 like-uri pe acele videoclipuri.

```

1 SELECT U.ume AS nume_utilizator,
2         V.titlu AS titlu_videoclip,
3         V.durata AS durata_videoclip,
4         COUNT(L.id_like) AS numar_like
5 FROM Utilizator U
6 JOIN Postari P ON U.id_utilizator = P.id_utilizator
7 JOIN Video V ON P.id_postari = V.id_postari
8 LEFT JOIN "LIKE" L ON V.id_postari = L.id_postari
9 WHERE V.durata > 100
10 GROUP BY U.ume, V.titlu, V.durata
11 HAVING COUNT(L.id_like) >= 2;

```

NUME_UTILIZATOR	TITLU_VIDEOCLIP	DURATA_VIDEOCLIP	NUMAR_LIKE
1 Maria Ionescu	Tutorial programare	300	2

The screenshot shows a SQL playground with a query editor and a results panel. The query is as follows:

```

1 SELECT U.ume AS nume_utilizator,
2       V.titlu AS titlu_videoclip,
3       V.durata AS durata_videoclip,
4       COUNT(L.id_like) AS numar_like
5 FROM Utilizator U
6 JOIN Postari P ON U.id_utilizator = P.id_utilizator
7 JOIN Video V ON P.id_postari = V.id_postari
8 LEFT JOIN "LIKE" L ON V.id_postari = L.id_postari
9 WHERE V.durata > 10
10 GROUP BY U.ume, V.titlu, V.durata
11 HAVING COUNT(L.id_like) >= 2;

```

The results panel shows one row of data:

NUME_UTILIZATOR	TITLU_VIDEOCLIP	DURATA_VIDEOCLIP	NUMAR LIKE
Maria Ionescu	Tutorial programare	300	2

3. Scrieți o cerere SQL care să obțină informații despre utilizatorii care fac parte din cel puțin 1 grup, au participat la cel puțin 1 eveniment și au postat cel puțin 1 postare. În plus, pentru postările acestor utilizatori, calculați media numărului de comentarii și like-uri pe postare.

```

1 SELECT
2     u.id_utilizator,
3     u.ume,
4     COUNT(DISTINCT mg.id_grup) AS numar_grupuri,
5     COUNT(DISTINCT pe.id_eveniment) AS numar_evenimente,
6     COUNT(p.id_postari) AS numar_postari,
7     AVG(c.numar_comentarii) AS media_comentarii,
8     AVG(l.numar_likes) AS media_likes
9 FROM
10    Utilizator u
11 JOIN
12    Membru_Grup mg ON u.id_utilizator = mg.id_utilizator
13 JOIN
14    Participant_Eveniment pe ON u.id_utilizator = pe.id_utilizator
15 JOIN
16    Postari p ON u.id_utilizator = p.id_utilizator
17 JOIN (
18     SELECT
19         id_postari,
20         COUNT(id_comentarii) AS numar_comentarii
21     FROM
22         Comentarii
23     GROUP BY

```

```

24         id_postari
25 ) c ON p.id_postari = c.id_postari
26 JOIN (
27     SELECT
28         id_postari,
29         COUNT(id_like) AS numar_likes
30     FROM
31         "LIKE"
32     GROUP BY
33         id_postari
34 ) l ON p.id_postari = l.id_postari
35 GROUP BY
36     u.id_utilizator,
37     u.nume
38 HAVING
39     COUNT(DISTINCT mg.id_grup) >= 1
40     AND COUNT(DISTINCT pe.id_eveniment) >= 1
41     AND COUNT(p.id_postari) >= 1;

```

ID_UTILIZATOR	NUME	NUMAR_GRUPURI	NUMAR_EVENIMENTE	NUMAR_POSTARI	MEDIA_COMENTARII	MEDIA_LIKES
1	Ion Popescu	1	1	1	1	1
2	Maria Ionescu	1	1	1	1	2
3	Ana Popa	1	1	1	1	1
4	Madelin Ioana	1	1	1	1	1
5	Elena Petrescu	1	1	1	1	1
6	Mihai Dumitru	1	1	1	1	1
7	Cristina Radu	1	1	1	1	1
8	Florin Stoica	1	1	1	1	1
9	Ioana Pavel	1	1	1	1	1


```
1 ✓ SELECT
2     u.id_utilizator,
3     u.nume,
4     COUNT(DISTINCT mg.id_grup) AS numar_grupuri,
5     COUNT(DISTINCT pe.id_eveniment) AS numar_evenimente,
6     COUNT(p.id_postari) AS numar_postari,
7     AVG(c.numar_comentarii) AS media_comentarii,
8     AVG(l.numar_likes) AS media_likes
9 FROM
10     Utilizator u
11 JOIN
12     Membru_Grup mg ON u.id_utilizator = mg.id_utilizator
13 JOIN
14     Participant_Eveniment pe ON u.id_utilizator = pe.id_utilizator
15 JOIN
16     Postari p ON u.id_utilizator = p.id_utilizator
17 JOIN (
18     SELECT
19         id_postari,
20         COUNT(id_comentarii) AS numar_comentarii
21     FROM
22         Comentarii
23     GROUP BY
24         id_postari
25 ) c ON p.id_postari = c.id_postari
26 JOIN (
27     SELECT
28         id_postari,
29         COUNT(id_like) AS numar_likes
```



```

23         id_utilizator,
24         COUNT(*) AS numar_postari
25     FROM
26         Postari
27     GROUP BY
28         id_utilizator
29 ) p ON u.id_utilizator = p.id_utilizator
30 LEFT JOIN (
31     SELECT
32         id_utilizator,
33         COUNT(*) AS numar_comentarii
34     FROM
35         Comentarii
36     GROUP BY
37         id_utilizator
38 ) c ON u.id_utilizator = c.id_utilizator
39 LEFT JOIN (
40     SELECT
41         id_utilizator,
42         COUNT(*) AS numar_likes
43     FROM
44         "LIKE"
45     GROUP BY
46         id_utilizator
47 ) l ON u.id_utilizator = l.id_utilizator
48 ORDER BY
49     u.data_inregistrare DESC,
50     u.numa ASC;

```

ID_UTILIZATOR	NUME	DATA_INREGISTRARE	NUMAR_POSTARI	NUMAR_COMENTARII	NUMAR_LIKES	TIP_UTILIZATOR
5	Madalina Ioana	2024-09-09	1	1	1	Pasiv
1	Ion Popescu	2024-05-05	1	1	2	Activ
8	Cristina Radu	2024-05-02	1	1	1	Activ
3	Stancu Rares	2023-10-01	1	1	1	Activ
9	Florin Stoica	2023-09-01	1	1	1	Activ
4	Ana Popa	2023-04-01	1	1	1	Activ
2	Maria Ionescu	2022-12-12	1	1	0	Activ
6	Elena Petrescu	2021-04-02	1	1	1	Activ
10	Ioana Pavel	2020-01-03	1	1	1	Activ
7	Mihai Dumitru	2019-11-11	1	1	1	Activ

```
1  ✓ SELECT
2      u.id_utilizator,
3      u.nume,
4      u.data_inregistrare,
5      NVL(p.numar_postari, 0) AS numar_postari,
6      NVL(c.numar_comentarii, 0) AS numar_comentarii,
7      NVL(l.numar_likes, 0) AS numar_likes,
8      DECODE(
9          SIGN(MONTHS_BETWEEN(SYSDATE, u.data_inregistrare)),
10         0, 'Nou',
11         1,
12         DECODE(
13             NVL(p.numar_postari, 0) + NVL(c.numar_comentarii, 0),
14             0, 'Pasiv',
15             'Activ'
16         ),
17         'Pasiv'
18     ) AS tip_utilizator
19 FROM
20     Utilizator u
21     LEFT JOIN (
22         SELECT
23             id_utilizator,
24             COUNT(*) AS numar_postari
25         FROM
26             Postari
27         GROUP BY
28             id_utilizator
29     ) p ON u.id_utilizator = p.id_utilizator
30     LEFT JOIN (
31         SELECT
32             id_utilizator,
```

```

33      COUNT(*) AS numar_comentarii
34    FROM
35      Comentarii
36    GROUP BY
37      id_utilizator
38  ) c ON u.id_utilizator = c.id_utilizator
39  LEFT JOIN (
40    SELECT
41      id_utilizator,
42      COUNT(*) AS numar_likes
43    FROM
44      "LIKE"
45    GROUP BY
46      id_utilizator
47  ) l ON u.id_utilizator = l.id_utilizator
48  ORDER BY
49    u.data_inregistrare DESC,
50    u.numa ASC;

```

ID_UTILIZATOR	NUME	DATA_INREGISTRARE	NUMAR_POSTARI	NUMAR_COMENTARII	NUMAR_LIKES	TIP_UTILIZATOR
5	Madalin Ioana	2024-09-09	1	1	1	Pasiv
1	Ion Popescu	2024-05-05	1	1	1	Activ
8	Cristina Radu	2024-05-02	1	1	1	Activ
3	Stancu Rares	2023-10-01	1	1	1	Activ
9	Florin Stoica	2023-09-01	1	1	1	Activ
4	Ana Popa	2023-04-01	1	1	1	Activ

5.Sa se extraga initiala Utilizatorilor, id_video,durata videoclipului (Scurt daca ; 60 secunde, Mediu 60-300secunde, Lung restul), Tipul reactiei(Apreciat daca tip_reactie = "Like", Neapreciat daca tip_reactie = "Dislike") numar like-uri, numar_comentarii despre utilizatorii care au postat cel puțin un video și care au primit cel puțin un like pe fiecare video postat. De asemenea, dorim să afișăm utilizatorii care au postat cel puțin un video în ultimele 30 de zile, dar nu au primit niciun like pe acele videoclipuri.

```

1  WITH
2    UtilizatoriCuVideo AS (
3      SELECT DISTINCT u.id_utilizator, u.numa
4      FROM Utilizator u
5      INNER JOIN Postari p ON u.id_utilizator = p.id_utilizator
6      INNER JOIN Video v ON p.id_postari = v.id_postari
7      WHERE v.data_upload >= CURRENT_DATE - INTERVAL '30' DAY
8    ),
9    StatisticiVideo AS (
10     SELECT
11       v.id_video,
12       COUNT(DISTINCT l.id_like) AS numar_likeuri,
13       COUNT(DISTINCT c.id_comentarii) AS numar_comentarii
14     FROM Video v
15     LEFT JOIN "LIKE" l ON v.id_postari = l.id_postari
16     LEFT JOIN Comentarii c ON v.id_postari = c.id_postari
17     GROUP BY v.id_video
18   )
19  SELECT
20    UPPER(SUBSTR(ucv.numa, 1, 1)) AS initiala_utilizator,
21    sv.id_video,

```

```

22     CASE
23         WHEN v.durata < 60 THEN 'Scurt'
24         WHEN v.durata BETWEEN 60 AND 300 THEN 'Mediu'
25         ELSE 'Lung'
26     END AS durata_video,
27     CASE
28         WHEN UPPER(r.tip_reactie) = 'LIKE' THEN 'Apreciat'
29         WHEN UPPER(r.tip_reactie) = 'DISLIKE' THEN 'Neapreciat'
30         ELSE r.TIP_REACTIE
31     END AS tip_reactie,
32     sv.numar_likeuri,
33     sv.numar_comentarii
34 FROM UtilizatoriCuVideo ucv
35 INNER JOIN Postari p ON ucv.id_utilizator = p.id_utilizator
36 INNER JOIN Video v ON p.id_postari = v.id_postari
37 LEFT JOIN StatisticiVideo sv ON v.id_video = sv.id_video
38 LEFT JOIN Reactie r ON v.id_postari = r.id_postari
39 WHERE
40     sv.numar_likeuri >= 1
41     AND NOT EXISTS (
42         SELECT 1
43         FROM UtilizatoriCuVideo ucv2
44         WHERE
45             ucv2.id_utilizator = ucv.id_utilizator
46             AND ucv2.id_utilizator NOT IN (
47                 SELECT l.id_utilizator
48                 FROM "LIKE" l
49                 WHERE l.id_postari = p.id_postari
50             )
51 );

```

	INITIALA_UTILIZATOR	ID_VIDEO	DURATA_VIDEO	TIP_REACTIE	NUMAR LIKEURI	NUMAR_COMENTARII
1	I		1 Scurt	Love	1	1
2	A		4 Mediu	Haha	1	1

```
1  ✓ WITH
2    UtilizatoriCuVideo AS (
3      SELECT DISTINCT u.id_utilizator, u.numa
4      FROM Utilizator u
5      INNER JOIN Postari p ON u.id_utilizator = p.id_utilizator
6      INNER JOIN Video v ON p.id_postari = v.id_postari
7      WHERE v.data_upload >= CURRENT_DATE - INTERVAL '30' DAY
8    ),
9    StatisticiVideo AS (
10     SELECT
11       v.id_video,
12       COUNT(DISTINCT l.id_like) AS numar_likeuri,
13       COUNT(DISTINCT c.id_comentarii) AS numar_comentarii
14     FROM Video v
15     LEFT JOIN "LIKE" l ON v.id_postari = l.id_postari
16     LEFT JOIN Comentarii c ON v.id_postari = c.id_postari
17     GROUP BY v.id_video
18   )
19  SELECT
20    UPPER(SUBSTR(ucv.numa, 1, 1)) AS initiala_utilizator,
21    sv.id_video,
22    CASE
23      WHEN v.durata < 60 THEN 'Scurt'
24      WHEN v.durata BETWEEN 60 AND 300 THEN 'Mediu'
25      ELSE 'Lung'
26    END AS durata_video,
27    CASE
28      WHEN UPPER(r.tip_reactie) = 'LIKE' THEN 'Apreciat'
29      WHEN UPPER(r.tip_reactie) = 'DISLIKE' THEN 'Neapreciat'
30      ELSE r.TIP_REACTIE
31    END AS tip_reactie,
32    sv.numar_likeuri,
33    sv.numar_comentarii
34  FROM UtilizatoriCuVideo ucv
35  INNER JOIN Postari p ON ucv.id_utilizator = p.id_utilizator
36  INNER JOIN Video v ON p.id_postari = v.id_postari
```

```

37 LEFT JOIN StatisticiVideo sv ON v.id_video = sv.id_video
38 LEFT JOIN Reactie r ON v.id_postari = r.id_postari
39 WHERE
40     sv.numar_likeuri >= 1
41     AND NOT EXISTS (
42         SELECT 1
43         FROM UtilizatoriCuVideo ucv2
44         WHERE
45             ucv2.id_utilizator = ucv.id_utilizator
46             AND ucv2.id_utilizator NOT IN (
47                 SELECT l.id_utilizator
48                 FROM "LIKE" l
49                 WHERE l.id_postari = p.id_postari
50             )
51     );
52

```

Output Result 4

	INITIALA_UTILIZATOR	ID_VIDEO	DURATA_VIDEO	TIP_REACTIE	NUMAR LIKEURI	NUMAR_COMENTARII
1	I	1	Scurt	Love	1	1
2	A	4	Mediu	Haha	1	1

6. Scrieți o interogare SQL pentru a obține un raport detaliat al bazei de date in care sa se arate informatii despre: numele utilizatorului, numărul total de postări făcute de utilizator, numărul total de comentarii făcute de utilizator, numărul total de reacții primite la postările utilizatorului, numărul total de like-uri primite la postările utilizatorului, numărul total de videoclipuri postate de utilizator, numărul total de poze postate de utilizator, numărul total de taguri adăugate la postările utilizatorului. In acest proces sa se utilizeze cel puțin un bloc de cerere (WITH) în interogare pentru a calcula aceste statistici.

```

1  WITH UserStats AS (
2      SELECT
3          U.id_utilizator,
4          U.num,
5          COUNT(DISTINCT P.id_postari) AS total_postari,
6          COUNT(DISTINCT C.id_comentarii) AS total_comentarii,
7          COUNT(DISTINCT R.id_reactie) AS total_reactii_primate,
8          COUNT(DISTINCT L.id_like) AS total_likes_primate,
9          COUNT(DISTINCT V.id_video) AS total_videoclipuri,
10         COUNT(DISTINCT Po.id_poze) AS total_poze,
11         COUNT(DISTINCT T.id_tag) AS total_taguri
12     FROM Utilizator U
13     LEFT JOIN Postari P ON U.id_utilizator = P.id_utilizator
14     LEFT JOIN Comentarii C ON U.id_utilizator = C.id_utilizator
15     LEFT JOIN Reactie R ON P.id_postari = R.id_postari
16     LEFT JOIN "LIKE" L ON P.id_postari = L.id_postari
17     LEFT JOIN Video V ON P.id_postari = V.id_postari
18     LEFT JOIN Poze Po ON P.id_postari = Po.id_postari
19     LEFT JOIN Tag T ON P.id_postari = T.id_postari

```



```

20     GROUP BY U.id_utilizator, U.numa
21 )
22 SELECT
23     US.numa,
24     US.total_postari,
25     US.total_comentarii,
26     US.total_reactii_prime,
27     US.total_likes_prime,
28     US.total_videoclipuri,
29     US.total_poze,
30     US.total_taguri
31 FROM UserStats US
32 ORDER BY US.numa;

```

	NUMA	TOTAL_POSTARI	TOTAL_COMENTARII	TOTAL_REACTII_PRIME	TOTAL_LIKES_PRIME	TOTAL_VIDEOCLIPURI	TOTAL_POZE	TOTAL_TAGURI
1	Ana Popa	1	1	1	1	1	1	1
2	Cristina Radu	1	1	1	1	1	1	1
3	Elena Petrescu	1	1	1	1	1	1	1
4	Florin Stoica	1	1	1	1	1	1	1
5	Ioana Pavel	1	1	1	1	1	1	1
6	Ion Popescu	1	1	1	1	1	1	1
7	Madalin Ioana	1	1	1	1	1	1	1
8	Maria Ionescu	1	1	1	2	1	1	1
9	Mihai Dumitru	1	1	1	1	1	1	1
10	Stancu Rares	1	1	1	0	1	1	1

```

1 WITH UserStats AS (
2     SELECT
3         U.id_utilizator,
4         U.numa,
5         COUNT(DISTINCT P.id_postari) AS total_postari,
6         COUNT(DISTINCT C.id_comentarii) AS total_comentarii,
7         COUNT(DISTINCT R.id_reactii) AS total_reactii_prime,
8         COUNT(DISTINCT L.id_likes) AS total_likes_prime,
9         COUNT(DISTINCT V.id_video) AS total_videoclipuri,
10        COUNT(DISTINCT Po.id_poze) AS total_poze,
11        COUNT(DISTINCT T.id_tag) AS total_taguri
12    FROM Utilizator U
13    LEFT JOIN Postari P ON U.id_utilizator = P.id_utilizator
14    LEFT JOIN Comentarii C ON U.id_utilizator = C.id_utilizator
15    LEFT JOIN Reactii R ON P.id_postari = R.id_postari
16    LEFT JOIN Likes L ON P.id_postari = L.id_postari
17    LEFT JOIN Video V ON P.id_postari = V.id_postari
18    LEFT JOIN Poze Po ON P.id_postari = Po.id_postari
19    LEFT JOIN Tag T ON P.id_postari = T.id_postari
20    GROUP BY U.id_utilizator, U.numa
21 )
22 SELECT
23     US.numa,
24     US.total_postari,
25     US.total_comentarii,
26     US.total_reactii_prime,
27     US.total_likes_prime,
28     US.total_videoclipuri,
29     US.total_poze,
30     US.total_taguri
31 FROM UserStats US
32 ORDER BY US.numa;

```

Output

Result 5

	NUMA	TOTAL_POSTARI	TOTAL_COMENTARII	TOTAL_REACTII_PRIME	TOTAL_LIKES_PRIME	TOTAL_VIDEOCLIPURI	TOTAL_POZE	TOTAL_TAGURI
1	Ana Popa	1	1	1	1	1	1	1

13 Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

13.1 Actualizarea coloanei "TIP_REACTIE" din tabela Reactie pentru utilizatorii care au reactionat cu "Wow" in "Omg".

```
1 UPDATE Reactie
2 SET tip_reactie = 'Omg'
3 WHERE tip_reactie = 'Wow';
```

```
1 ✓ ▾ UPDATE Reactie
2     SET tip_reactie = 'Omg'
3     WHERE tip_reactie = 'Wow';
4
```

```
[2024-06-07 16:21:47] Connected
UTILIZATOR> alter session set current_schema = UTILIZATOR
[2024-06-07 16:21:47] completed in 1 ms
UTILIZATOR> UPDATE Reactie
              SET tip_reactie = 'Omg'
              WHERE tip_reactie = 'Wow'
[2024-06-07 16:21:47] 2 rows affected in 25 ms
```

13.2 Stergerea înregistrărilor din tabela Postari pentru comentariile care se referă la postări mai vechi de 2023

```
1 DELETE FROM Comentarii
2 WHERE id_postari IN (
3     SELECT id_postari
4     FROM Postari
5     WHERE data_postarii < DATE '2023-01-01'
6 );
```


13.3 Ștergere: Eliminarea comentariilor postate înainte de anul 2022 pentru postările care au primit reacții de tip "Omg"

```
1 DELETE FROM Comentarii
2 WHERE data_comentariu < DATE '2022-01-01'
3 AND id_postari IN (
4     SELECT R.id_postari
5     FROM Reactie R
6     WHERE R.tip_reactie = 'Omg'
7 );
8
9 );
```

MESAJ [project]

Playground ▾

```
[2024-06-07 16:40:01] completed in 5 ms
```

13.4 Actualizare: Schimbarea parolelor utilizatorilor care au creat postări video mai scurte de 40 minute.

```
1 UPDATE Utilizator
2 SET parola = 'parola_noua'
3 WHERE id_utilizator IN (
4     SELECT U.id_utilizator
5     FROM Utilizator U
6     JOIN Postari P ON U.id_utilizator = P.id_utilizator
7     JOIN Video V ON P.id_postari = V.id_postari
8     WHERE V.durata < 40)
```


Tx: Manual

Playground

1

✓

UPDATE Utilizator

2

SET parola = 'parola_nova'

3

WHERE id_utilizator IN (

4

✓

SELECT U.id_utilizator

5

FROM Utilizator U

6

JOIN Postari P ON U.id_utilizator = P.id_utilizator

7

JOIN Video V ON P.id_postari = V.id_postari

8

WHERE V.durata < 40)

9

[2024-06-07 16:42:40] 1 row affected in 7 ms

UTILIZATOR> UPDATE Utilizator

SET parola = 'parola_nova'

WHERE id_utilizator IN (

SELECT U.id_utilizator

FROM Utilizator U

JOIN Postari P ON U.id_utilizator = P.id_utilizator

JOIN Video V ON P.id_postari = V.id_postari

WHERE V.durata < 40)

[2024-06-07 16:43:13] 1 row affected in 6 ms