

FACULTATEA DE MATEMATICA SI INFORMATICA  
UNIVERSITATEA DIN BUCURESTI

**PROIECT APLICATII PROFESIONALE IN BAZE DE DATE**

**Profesor indrumator**

Lect.dr. Velcescu Letitia

**Autori**

Cernea Teodora

Bogdan Mihai Timofte

Bucuresti, 2015

# CUPRINS

<b>1. Descrierea modelului real</b>	<b>3</b>
<b>2. Constrangeri</b>	<b>4</b>
<b>3. Descrierea entitatilor, relatiilor, atriburelor</b>	<b>4</b>
3.1. Entitati	4
3.2. Relatii	6
3.3. Atribute	10
<b>4. Scheme relationale</b>	<b>17</b>
<b>5. Normalizare</b>	<b>19</b>
<b>6. Denormalizare</b>	<b>20</b>
<b>7. Crearea bazei de date</b>	<b>21</b>
<b>8. Ce se doreste de la model</b>	<b>29</b>
<b>9. Diagrama conceptuala a modelului</b>	<b>29</b>

# Modelarea unei retele de socializare

## 1. Descrierea modelului real

Proiectul invoca crearea unui model responsabil cu reprezentarea informatiilor referitoare la gestiunea unei retele de socializare.

Principalul actor al modelului nostru relational este utilizatorul (USER) ce are posibilitatea de a realiza diferite tipuri de actiuni in sistem. In primul rand, acesta este caracterizat de un profil (PROFILE) pentru care poate configura adresa, orasul (CITY), judetul (COUNTY), precum si tara (COUNTRY). De asemenea, i se acorda dreptul de a-si alege limba (LANGUAGE).

Interactiunea unui utilizator cu alti utilizatori da nastere conceptului de lista de prieteni (FRIEND\_LIST). Aceasta schiteaza relatia de prietenie virtuala dintre doua sau mai multe persoane, ce se realizeaza prin intermediul cererilor de prietenie. Mai mult de atat, utilizatorii aplicatiei pot comunica prin intermediul mesajelor (MESSAGE).

Datele cu caracter personal (profil, adresa, localitate, judet, tara), precum si pozele sau albumele unui utilizator sunt protejate prin optiunea (PRIVACY), ramanand la decizia acestuia de a le face publice, private sau vizibile doar de catre prieteni.

Dintre actiunile pe care le poate intreprinde un utilizator in aplicatie, enumeram :

- Publicarea unei stari (STATUS);
- Crearea si publicarea unui album (ALBUM);
- Incarcarea si postarea unei poze (PICTURE);
- Adaugarea, editarea, stergerea unui comentariu (COMMENT) – adaugat la o poza (COMMENT\_PICTURE), la un album (COMMENT\_ALBUM) sau la un status (COMMENT\_STATUS);
- Aprecierea (LIKE) – a unei poze (LIKE\_PICTURE), a unui album (LIKE\_ALBUM), a unui status (LIKE\_STATUS) sau a unui comentariu (LIKE\_COMMENT) postate de alti utilizatori ce fac parte din lista de prieteni.
- Crearea si/sau participarea (PARTICIPATE\_EVENT) la un eveniment (EVENT).

Fiecare utilizator este la curent cu actiunile realizate de prietenii sai prin intermediul stirilor (FEED). In cadrul stirilor-ului apar inregistrari legate de pozele (FEED\_PICTURE), albumele (FEED\_ALBUM) create, precum si de modificarile de status-uri (FEED\_STATUS) existente. Mai mult, utilizatorii sunt instiintati si de evenimentele create (FEED\_EVENT) de catre alti prieteni.

Aplicatia pune la dispozitie si un modul de notificari (NOTIFICATION) prin care fiecare utilizator este instiintat de interactiunile directe ale altor utilizatori. Acestea se pot realiza prin intermediul comentariilor (NOTIFICATION\_COMMENT) sau like-urilor

(NOTIFICATION\_LIKE) si se reflecta asupra pozelor, albumelor si asupra statusurilor utilizatorului. Acceptarea unei cereri de prietenie trimise e redada de o notificare de aprobare a prieteniei (NOTIFICATION\_FRIEND).

## 2. Constrangeri

- Un user nu se poate imprieteni cu el insusi
- Un user are un singur profil.
- Un utilizator poate crea unul sau mai multe albume.
- Un album contine una sau mai multe poze.
- Mesajele dintre utilizatorii care nu sunt prieteni vor fi marcate ca fiind SPAM.

## 3. Descrierea entitatilor, relatiilor si atributelor

In urmatoarea sectiune vor fi prezentate entitatile, relatiile si attributele ce duc la modelarea unei retele de socializare ce are functionalitatile mai sus amintite.

### 3.1. Entitati

**USER** = Cheie primara : user\_id. Entitatea modeleaza si inregistreaza in aplicatie personajul principal : utilizatorul aplicatiei. Acesta este caracterizat de cateva date personale.

**PROFILE** = Cheie primara : profile\_id. Profilul unui utilizator contine informatii despre preferintele acestuia, educatia si obiceiurile sale.

**CITY** = Cheie primara : city\_id. Identifica orasul in care locuieste utilizatorul.

**COUNTY** = Cheie primara : county\_id. Judet sau sector in care locuieste utilizatorul.

**COUNTRY** = Cheie primara : country\_id. Tara in care locuieste utilizatorul.

**LANGUAGE** = Cheie primara : language\_id. Limba de circulatie pe care utilizatorul o seteaza.

**FRIEND\_LIST** = Cheie primara : friend\_list\_id. Lista ce retine relatiile de prietenie dintre doi sau mai multi utilizatori.

**MESSAGE** = Cheie primara : message\_id. Reprezinta mesajele ce sunt transmise intre utilizatorii ce sunt prieteni.

**STATUS** = Cheie primara : status\_id. Starea de spirit, gandurile pe care un utilizator le impartaseste virtual cu prietenii sai.

**PRIVACY** = Cheie primara : primary\_id. Optiune disponibila utilizatorului prin care acesta isi poate configura setarile privind publicul tinta al postarilor sale: public, privat, etc.

**ALBUM** = Cheie primara : album\_id. Albumul pe care il creeaza utilizatorul si care contine pozele incarcate de acesta.

**PICTURE** = Cheie primara : picture\_id. Poza pe care utilizatorul o poate incarca in mediul online impreuna cu o descriere.

**COMMENT** = Cheie primara : comment\_id. Mesaj pe care un utilizator il poate lasa - public la poza, albumul sau statusul altui utilizator.

**COMMENT\_PICTURE** = Cheie primara : comment\_picture\_id. Comentariul pe care un utilizator il poate lasa la poza incarcata de alt utilizator.

**COMMENT\_ALBUM** = Cheie primara : comment\_album\_id. Comentariul pe care un utilizator il poate lasa la albumul creat de alt utilizator.

**COMMENT\_STATUS** = Cheie primara : comment\_status\_id. Comentariul pe care un utilizator il poate lasa la statusul unui utilizator.

**LIKE** = Cheie primara : like\_id. Reprezinta semnul de apreciere al unui utilizator fata de postarile altui utilizator.

**LIKE\_PICTURE** = Cheie primara : like\_picture\_id. Aprecierea unei poze.

**LIKE\_ALBUM** = Cheie primara : like\_album\_id. Aprecierea unui album.

**LIKE\_STATUS** = Cheie primara : like\_status\_id. Aprecierea unui status.

**LIKE\_COMMENT** = Cheie primara : like\_comment\_id. Aprecierea unui status.

**EVENT** = Cheie primara : event\_id. Identifica evenimentele de natura sociala, educationala ce exista in sistem.

**PARTICIPATE\_EVENT** = Cheie primara : participate\_event\_id. Participarea la un eveniment.

**FEED** = Cheie primara : feed\_id. Noutatile postate de alti utilizatori pe care un utilizator le vede.

**FEED\_PICTURE** = Cheie primara : feed\_picture\_id. Pozele noi pe care le-au incarcat alti utilizatori si pe care le vede un utilizator.

**FEED\_ALBUM** = Cheie primara : feed\_album\_id. Albumele noi create de catre alti utilizatori pe care le vede un utilizator.

**FEED\_STATUS** = Cheie primara : feed\_status\_id. Statusurile noi pe care le creeaza sau modifica alti utilizatori si pe care le vede un utilizator.

**FEED\_EVENT** = Cheie primara : feed\_event\_id. Evenimentele noi create si participarea altor utilizatori la acestea pe care un utilizator le vede.

**NOTIFICATION** = Cheie primara: notification\_id. Notificarile ce apar pe contul unui utilizator.

**NOTIFICATION\_COMMENT** – Cheie primara : notification\_comment\_id. Notificarea pe care o primeste un utilizator atunci cand prietenii sai ii lasa comentarii la postari (albume, poze, statusuri).

**NOTIFICATION\_LIKE** – Cheie primara : notification\_like\_id. Notificarea pe care o primeste un utilizator atunci cand prietenii sai apreciaza una dintre postarile sale (albume, poze, statusuri, comentarii).

**NOTIFICATION\_FRIEND** – Cheie primara : notification\_friend\_id. Notificarea pe care o primeste un utilizator atunci cand un alt utilizator ii trimite sau accepta o cerere de prietenie.

### 3.2. Relatii

**USER\_are\_PROFILE**: relatie ce leaga entitatile USER si PROFILE. Cardinalitate minima: 1:1. Cardinalitate maxima: 1:1.

**USER\_FRIEND\_LIST**: relatie ce leaga entitatile USER si FRIEND\_LIST. Cardinalitate minima: 1:0. Cardinalitate maxima: 1:n.

**USER\_trimite\_MESSAGE**: relatie ce leaga entitatile USER si MESSAGE. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**PROFILE\_are\_CITY**: relatie ce leaga entitatile PROFILE si CITY. Cardinalitatea minima: 0:0. Cardinalitatea maxima: n:1.

**CITY\_apartine\_COUNTY**: relatie ce leaga entitatile CITY si COUNTY. Cardinalitatea minima: 1:1. Cardinalitatea maxima: n:1.

**COUNTY\_apartine\_COUNTRY**: relatie ce leaga entitatile COUNTY si COUNTRY. Cardinalitatea minima: 1:1. Cardinalitatea maxima: n:1.

**USER\_seteaza\_LANGUAGE**: relatie ce leaga entitatile USER si LANGUAGE. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_posteaza\_STATUS**: relatie ce leaga entitatile USER si STATUS. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**USER\_creeaza\_ALBUM**: relatie ce leaga entitatile USER si ALBUM. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**ALBUM\_contine\_PICTURE:** relatie ce leaga entitatile USER si PICTURE. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**USER\_LIKE\_PICTURE:** relatie ce leaga entitatile USER si LIKE\_PICTURE. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**LIKE\_PICTURE\_PICTURE:** relatie ce leaga entitatile LIKE\_PICTURE si PICTURE. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_LIKE\_ALBUM:** relatie ce leaga entitatile USER si LIKE\_ALBUM. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**LIKE\_ALBUM\_ALBUM:** relatie ce leaga entitatile LIKE\_ALBUM si ALBUM. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_LIKE\_STATUS:** relatie ce leaga entitatile USER si LIKE\_STATUS. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**LIKE\_STATUS\_STATUS:** relatie ce leaga entitatile LIKE\_STATUS si STATUS. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_LIKE\_COMMENT:** relatie ce leaga entitatile USER si LIKE\_COMMENT. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**LIKE\_COMMENT\_COMMENT:** relatie ce leaga entitatile LIKE\_COMMENT si COMMENT. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**LIKE\_LIKE\_PICTURE:** relatie ce leaga entitatile LIKE si LIKE\_PICTURE. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**LIKE\_LIKE\_ALBUM:** relatie ce leaga entitatile LIKE si LIKE\_ALBUM. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**LIKE\_LIKE\_STATUS:** relatie ce leaga entitatile LIKE si LIKE\_STATUS. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**LIKE\_LIKE\_COMMENT:** relatie ce leaga entitatile LIKE si LIKE\_COMMENT. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**USER\_COMMENT\_PICTURE:** relatie ce leaga entitatile USER si COMMENT\_PICTURE. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**COMMENT\_PICTURE\_PICTURE:** relatie ce leaga entitatile COMMENT\_PICTURE si PICTURE. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_COMMENT\_ALBUM:** relatie ce leaga entitatile USER si COMMENT\_ALBUM. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**COMMENT\_ALBUM\_ALBUM:** relatie ce leaga entitatile COMMENT\_ALBUM si ALBUM.  
Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_COMMENT\_STATUS:** relatie ce leaga entitatile USER si COMMENT\_STATUS.  
Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**COMMENT\_STATUS\_STATUS:** relatie ce leaga entitatile COMMENT\_STATUS si STATUS.  
Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**COMMENT\_COMMENT\_PICTURE:** relatie ce leaga entitatile COMMENT si  
COMMENT\_PICTURE. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**COMMENT\_COMMENT\_ALBUM:** relatie ce leaga entitatile COMMENT si  
COMMENT\_ALBUM. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**COMMENT\_COMMENT\_STATUS:** relatie ce leaga entitatile COMMENT si  
COMMENT\_STATUS. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**PICTURE\_are\_PRIVACY:** relatie ce leaga entitatile PICTURE si PRIVACY. Cardinalitatea  
minima: 1:1. Cardinalitatea maxima: 1:1.

**ALBUM\_are\_PRIVACY:** relatie ce leaga entitatile ALBUM si PRIVACY. Cardinalitatea minima:  
1:1. Cardinalitatea maxima: 1:1.

**STATUS\_are\_PRIVACY:** relatie ce leaga entitatile STATUS si PRIVACY. Cardinalitatea minima:  
1:1. Cardinalitatea maxima: 1:1.

**PROFILE\_are\_PRIVACY:** relatie ce leaga entitatile PROFILE si PRIVACY. Cardinalitatea minima:  
1:1. Cardinalitatea maxima: 1:1.

**CITY\_are\_PRIVACY:** relatie ce leaga entitatile CITY si PRIVACY. Cardinalitatea minima: 1:1.  
Cardinalitatea maxima: 1:1.

**COUNTY\_are\_PRIVACY:** relatie ce leaga entitatile COUNTY si PRIVACY. Cardinalitatea  
minima: 1:1. Cardinalitatea maxima: 1:1.

**COUNTRY\_are\_PRIVACY:** relatie ce leaga entitatile COUNTRY si PRIVACY. Cardinalitatea  
minima: 1:1. Cardinalitatea maxima: 1:1.

**COUNTRY\_are\_PRIVACY:** relatie ce leaga entitatile COUNTRY si PRIVACY. Cardinalitatea  
minima: 1:1. Cardinalitatea maxima: 1:1.

**USER\_creeaza\_EVENT:** relatie ce leaga entitatile USER si EVENT. Cardinalitatea minima: 1:0.  
Cardinalitatea maxima: 1:n.

**USER\_PARTICIPATE\_EVENT:** relatie ce leaga entitatile USER si PARTICIPATE\_EVENT.  
Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.



**PARTICIPATE\_EVENT\_EVENT**: relatie ce leaga entitatile PARTICIPATE\_EVENT si EVENT.  
Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**EVENT\_are\_CITY**: relatie ce leaga entitatile EVENT si CITY. Cardinalitatea minima: 0:1.  
Cardinalitatea maxima: n:1.

**USER\_vizualizeaza\_FEED\_PICTURE**: relatie ce leaga entitatile USER si FEED\_PICTURE.  
Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**FEED\_PICTURE\_contine\_PICTURE**: relatie ce leaga entitatile FEED\_PICTURE si PICTURE.  
Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_vizualizeaza\_FEED\_ALBUM**: relatie ce leaga entitatile USER si FEED\_ALBUM.  
Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**FEED\_ALBUM\_contine\_ALBUM**: relatie ce leaga entitatile FEED\_ALBUM si ALBUM.  
Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_vizualizeaza\_FEED\_STATUS**: relatie ce leaga entitatile USER si FEED\_STATUS.  
Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**FEED\_STATUS\_contine\_STATUS**: relatie ce leaga entitatile FEED\_STATUS si STATUS.  
Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_vizualizeaza\_FEED\_EVENT**: relatie ce leaga entitatile USER si FEED\_EVENT.  
Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**FEED\_EVENT\_contine\_EVENT**: relatie ce leaga entitatile FEED\_EVENT si EVENT.  
Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**FEED\_FEED\_PICTURE**: relatie ce leaga entitatile FEED si FEED\_PICTURE. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**FEED\_FEED\_ALBUM**: relatie ce leaga entitatile FEED si FEED\_ALBUM. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**FEED\_FEED\_STATUS**: relatie ce leaga entitatile FEED si FEED\_STATUS. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**FEED\_FEED\_EVENT**: relatie ce leaga entitatile FEED si FEED\_EVENT. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**USER\_primeste\_NOTIFICATION\_LIKE**: relatie ce leaga entitatile USER si NOTIFICATION\_LIKE.  
Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**NOTIFICATION\_LIKE\_contine\_LIKE**: relatie ce leaga entitatile NOTIFICATION\_LIKE si LIKE.  
Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**USER\_primeste\_NOTIFICATION\_COMMENT:** relatie ce leaga entitatile USER si NOTIFICATION\_COMMENT. Cardinalitatea minima: 1:0. Cardinalitatea maxima: 1:n.

**NOTIFICATION\_COMMENT\_contine\_COMMENT:** relatie ce leaga entitatile NOTIFICATION\_COMMENT si COMMENT. Cardinalitatea minima: 0:1. Cardinalitatea maxima: n:1.

**NOTIFICATION\_FRIEND\_FRIEND\_LIST:** relatie ce leaga entitatile NOTIFICATION\_FRIEND si FRIEND\_LIST. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**NOTIFICATION\_NOTIFICATION\_LIKE:** relatie ce leaga entitatile NOTIFICATION si NOTIFICATION\_LIKE. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**NOTIFICATION\_NOTIFICATION\_COMMENT:** relatie ce leaga entitatile NOTIFICATION si NOTIFICATION\_COMMENT. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

**NOTIFICATION\_NOTIFICATION\_FRIEND:** relatie ce leaga entitatile NOTIFICATION si NOTIFICATION\_FRIEND. Cardinalitatea minima: 0:0. Cardinalitatea maxima: 1:1.

### 3.3. Atribute

#### USER

Nume	Tip	Descriere
user_id	number(6)	Cheie primara
language_id	number(6)	Cheie externa, facand legatura cu limba pe care utilizatorul o seteaza
username	nvarchar2(255)	Nume ales de utilizator
password	nvarchar2(255)	Parola aleasa de utilizator
first_name	nvarchar2(255)	Prenumele utilizatorului
last_name	nvarchar2(255)	Numele utilizatorului
email	nvarchar2(255)	E-mailul utilizatorului
active	nchar(1)	Starea contului: activ sau inactiv
created_at	date	Data crearii contului utilizatorului

#### PROFILE

Nume	Tip	Descriere
profile_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul.
privacy_id	number(6)	Cheie externa, realizand legatura cu PRIVACY
city_id	number(6)	Cheie externa, realizand legatura cu orasul
date_of_birth	date	Data de nastere a utilizatorului
about_me	nvarchar2(255)	Descrierea utilizatorului
relationship	nvarchar2(255)	Starea civila

phone	nvarchar2(255)	Numarul de telefon
interests	nvarchar2(255)	Interesele utilizatorului
education	nvarchar2(255)	Educatia utilizatorului
hobbies	nvarchar2(255)	Pasiunile, obiceiurile utilizatorului
fav_movies	nvarchar2(255)	Filmele preferate de utilizator
fav_artists	nvarchar2(255)	Artistii preferati de utilizator
fav_books	nvarchar2(255)	Cartile preferate de utilizator
address	nvarchar2(255)	Adresa utilizatorului

#### CITY

Nume	Tip	Descriere
city_id	number(6)	Cheie primara
county_id	number(6)	Cheie externa, realizand legatura cu judetul
privacy_id	number(6)	Cheie externa, realizand legatura cu PRIVACY
name	nvarchar2(255)	Numele orasului

#### COUNTY

Nume	Tip	Descriere
county_id	number(6)	Cheie primara
country_id	number(6)	Cheie externa, realizand legatura cu tara
privacy_id	number(6)	Cheie externa, realizand legatura cu PRIVACY
name	nvarchar2(255)	Numele judetului

#### COUNTRY

Nume	Tip	Descriere
country_id	number(6)	Cheie primara
privacy_id	number(6)	Cheie externa, realizand legatura cu PRIVACY
name	nvarchar2(255)	Numele tarii

#### LANGUAGE

Nume	Tip	Descriere
language_id	number(6)	Cheie primara
name	nvarchar2(255)	Numele limbii de circulatie

#### FRIEND\_LIST

Nume	Tip	Descriere
friend_list_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul
friend_user_id	number(6)	Cheie externa, realizand legatura cu prietenul utilizatorului
state	nchar(1)	Starea relatiei de prietenie: acceptata sau nu
created_at	date	Data trimiterii cererii de prietenie de catre un utilizator

#### MESSAGE

Nume	Tip	Descriere
------	-----	-----------

message_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care trimite/ primeste mesajul
friend_user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care primeste/ trimite mesajul
message_content	nvarchar2(255)	Continutul mesajului
created_at	date	Data crearii mesajului
is_read	nchar(1)	Starea mesajului: citit sau nu
is_spam	nchar(1)	Tipul mesajului: spam sau nu

## STATUS

Nume	Tip	Descriere
status_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul
privacy_id	number(6)	Cheie externa, realizand legatura cu PRIVACY
status_content	nvarchar2(255)	Continutul statusului
created_at	date	Data crearii statusului
updated_at	date	Data editarii/ modificarii statusului
numberOfLikes	int	Numarul de like-uri detinute de un status

## PRIVACY

Nume	Tip	Descriere
privacy_id	number(6)	Cheie primara
privacy_type	nvarchar2(255)	Tip: public, vizualizat doar de mine (only me), vizualizat doar de prietenii mei (only friends)

## PICTURE

Nume	Tip	Descriere
picture_id	number(6)	Cheie primara
album_id	number(6)	Cheie externa, realizand legatura cu albumul din care face parte poza
privacy_id	number(6)	Cheie externa, realizand legatura cu PRIVACY
description	nvarchar2(255)	Descrierea pozei
created_at	date	Data incarcarii pozei
updated_at	date	Data editarii, modificarii descrierii pozei
is_album_cover	nchar(1)	Daca este sau nu coperta albumului din care face parte
is_profile_picture	nchar(1)	Daca este sau nu poza de profil
numberOfLikes	int	Numarul de like-uri detinute de o poza

## ALBUM

Nume	Tip	Descriere
album_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul
privacy_id	number(6)	Cheie externa, realizand legatura cu PRIVACY
name	nvarchar2(255)	Numele albumului

description	nvarchar2(255)	Descrierea albumului
created_at	date	Data crearii albumului
updated_at	date	Data editarii, modificarii descrierii sau numelui albumului
numberOfLikes	int	Numarul de like-uri detinute de un album

#### COMMENT

Nume	Tip	Descriere
comment_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care a postat comentariul
comment_type_id	number(6)	Cheie externa, realizand legatura cu tipul comentariului
comment_type	nvarchar2(255)	Tipul comentariului

#### COMMENT\_PICTURE

Nume	Tip	Descriere
comment_picture_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care a postat comentariul
picture_id	number(6)	Cheie externa, realizand legatura cu poza la care a comentat utilizatorul
comment_content	nvarchar2(255)	Continutul comentariului
created_at	date	Data postarii comentariului
updated_at	date	Data editarii, modificarii comentariului
numberOfLikes	int	Numarul de like-uri detinute de un comentariu

#### COMMENT\_ALBUM

Nume	Tip	Descriere
comment_album_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care a postat comentariul
album_id	number(6)	Cheie externa, realizand legatura cu albumul la care a comentat utilizatorul
comment_content	nvarchar2(255)	Continutul comentariului
created_at	date	Data postarii comentariului
updated_at	date	Data editarii, modificarii comentariului
numberOfLikes	int	Numarul de like-uri detinute de un comentariu

#### COMMENT\_STATUS

Nume	Tip	Descriere
comment_status_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care a postat comentariul
status_id	number(6)	Cheie externa, realizand legatura cu statusul la care a comentat utilizatorul

comment_content	nvarchar2(255)	Continutul comentariului
created_at	date	Data postarii comentariului
updated_at	date	Data editarii, modificarii comentariului
numberOfLikes	int	Numarul de like-uri detinute de un comentariu

#### LIKE

Nume	Tip	Descriere
like_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care a dat LIKE
like_type_id	number(6)	Cheie externa, realizand legatura cu tipul de like
like_type	nvarchar(2)	Tipul de like

#### LIKE\_PICTURE

Nume	Tip	Descriere
like_picture_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care a dat LIKE
picture_id	number(6)	Cheie externa, realizand legatura cu poza pe care a apreciat-o utilizatorul
created_at	date	Data la care un utilizator a dat LIKE

#### LIKE\_ALBUM

Nume	Tip	Descriere
like_album_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care a dat LIKE
album_id	number(6)	Cheie externa, realizand legatura cu albumul pe care l-a apreciat utilizatorul
created_at	date	Data la care un utilizator a dat LIKE

#### LIKE\_STATUS

Nume	Tip	Descriere
like_status_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care a dat LIKE
status_id	number(6)	Cheie externa, realizand legatura cu statusul pe care l-a apreciat utilizatorul
created_at	date	Data la care un utilizator a dat LIKE

#### LIKE\_COMMENT

Nume	Tip	Descriere
like_comment_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care a dat LIKE

comment_id	number(6)	Cheie externa, realizand legatura cu comentariul pe care l-a apreciat utilizatorul
created_at	date	Data la care un utilizator a dat LIKE

#### EVENT

Nume	Tip	Descriere
event_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care a creat evenimentul
city_id	number(6)	Cheie externa, realizand legatura cu orasul evenimentului
name	nvarchar2(255)	Numele evenimentului
description	nvarchar2(255)	Descrierea evenimentului
address	nvarchar2(255)	Adresa evenimentului
date	date	Data la care are loc evenimentul
created_at	date	Data crearii evenimentului
updated_at	date	Data modificarii evenimentului

#### PARTICIPATE\_EVENT

Nume	Tip	Descriere
participate_event_id	number(6)	Cheie primara
event_id	number(6)	Cheie externa, realizand legatura cu un eveniment
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care participa la un eveniment

#### FEED

Nume	Tip	Descriere
feed_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul
feed_type_id	number(6)	Cheie externa, realizand legatura cu tipul FEED-ului
feed_type	nvarchar2(255)	Tipul Feed-ului

#### FEED\_PICTURE

Nume	Tip	Descriere
feed_picture_id	number(6)	Cheie primara
picture_id	number(6)	Cheie externa, realizand legatura cu poza nou postata de un utilizator
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care vizualizeaza FEED-uri

#### FEED\_ALBUM

Nume	Tip	Descriere
feed_album_id	number(6)	Cheie primara
album_id	number(6)	Cheie externa, realizand legatura cu albumul nou postat de un utilizator

user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care vizualizeaza FEED-uri
---------	-----------	--

#### FEED\_STATUS

Nume	Tip	Descriere
feed_status_id	number(6)	Cheie primara
status_id	number(6)	Cheie externa, realizand legatura cu statusul nou postat de un utilizator
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care vizualizeaza FEED-uri

#### FEED\_EVENT

Nume	Tip	Descriere
feed_event_id	number(6)	Cheie primara
event_id	number(6)	Cheie externa, realizand legatura cu evenimentul nou creat de un utilizator
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care vizualizeaza FEED-uri

#### NOTIFICATION

Nume	Tip	Descriere
notification_id	number(6)	Cheie primara
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care posteaza comentarii
notification_type_id	number(6)	Cheie externa, realizand legatura cu tipul notificarii
notification_type	nvarchar2(255)	Tipul notificarii

#### NOTIFICATION\_COMMENT

Nume	Tip	Descriere
notification_comment_id	number(6)	Cheie primara
comment_id	number(6)	Cheie externa, realizand legatura cu comentariul nou ce a fost adaugat la postarile unui utilizator.
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care posteaza comentarii
content	nvarchar2(255)	Continutul notificarii

#### NOTIFICATION\_LIKE

Nume	Tip	Descriere
notification_like_id	number(6)	Cheie primara
like_id	number(6)	Cheie externa, realizand legatura cu LIKE-ul dat de un utilizator la postarile altui utilizator.
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care da LIKE-uri
content	nvarchar2(255)	Continutul notificarii



## NOTIFICATION\_FRIEND

Nume	Tip	Descriere
notification_friend_id	number(6)	Cheie primara
friend_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care a trimis cererea de prietenie.
user_id	number(6)	Cheie externa, realizand legatura cu utilizatorul care accepta cereri de prietenie
content	nvarchar2(255)	Continutul notificarii

## 4. Scheme relationale

**USER** (#user\_id, language\_id, username, password, first\_name, last\_name, email, active, created\_at)

**PROFILE** (#profile\_id, user\_id, privacy\_id, city\_id, date\_of\_birth, about\_me, relationship, phone, interests, education, hobbies, fav\_movies, fav\_artists, fav\_books, address)

**CITY** (#city\_id, county\_id, privacy\_id, name)

**COUNTY** (#county\_id, country\_id, privacy\_id, name)

**COUNTRY** (#country\_id, privacy\_id, name)

**LANGUAGE** (#language\_id, name)

**FRIEND\_LIST** (#friend\_list\_id, user\_id, friend\_user\_id, state, created\_at)

**MESSAGE** (#message\_id, user\_id, friend\_user\_id, message\_content, created\_at, is\_read, is\_spam)

**STATUS** (#status\_id, user\_id, privacy\_id, status\_content, created\_at, updated\_at, numberOfLikes)

**PRIVACY** (#privacy\_id, privacy\_type)

**ALBUM** (#album\_id, user\_id, privacy\_id, name, description, created\_at, updated\_at, numberOfLikes)

**PICTURE** (#picture\_id, album\_id, privacy\_id, description, created\_at, updated\_at, is\_album\_cover, is\_profile\_picture, numberOfLikes)

**COMMENT\_STATUS** (#comment\_status\_id, user\_id, status\_id, comment\_content, created\_at, updated\_at, numberOfLikes)

**COMMENT\_ALBUM** (#comment\_album\_id, user\_id, album\_id, comment\_content, created\_at, updated\_at, numberOfLikes)

**COMMENT\_PICTURE** (#comment\_picture\_id, user\_id, picture\_id, comment\_content, created\_at, updated\_at, numberOfLikes)

**COMMENT** (#comment\_id, user\_id, comment\_type\_id, comment\_type)

**LIKE\_STATUS** (#like\_status\_id, user\_id, status\_id, created\_at)

**LIKE\_COMMENT** (#like\_comment\_id, user\_id, comment\_id, created\_at)

**LIKE\_ALBUM** (#like\_album\_id, user\_id, album\_id, created\_at)

**LIKE\_PICTURE** (#like\_picture\_id, user\_id, picture\_id, created\_at)

**LIKE** (#like\_id, user\_id, like\_type\_id, like\_type)

**EVENT** (#event\_id, #user\_id, city\_id, name, description, address, date, created\_at, updated\_at)

**PARTICIPATE\_EVENT** (#participate\_event\_id, event\_id, user\_id)

**FEED\_STATUS** (#feed\_status\_id, status\_id, user\_id)

**FEED\_ALBUM** (#feed\_album\_id, album\_id, user\_id)

**FEED\_PICTURE** (#feed\_picture\_id, picture\_id, user\_id)

**FEED\_EVENT** (#feed\_event\_id, event\_id, user\_id)

**FEED** (#feed\_id, user\_id, feed\_type\_id, feed\_type)

**NOTIFICATION\_COMMENT** (#notification\_comment\_id, comment\_id, user\_id, content)

**NOTIFICATION\_LIKE** (#notification\_like\_id, like\_id, user\_id, content)

**NOTIFICATION\_FRIEND** (#notification\_friend\_id, friend\_id, user\_id, content)

**NOTIFICATION** (#notification\_id, user\_id, notification\_type\_id, notification\_type)

## 5. Normalizare

Pentru a exemplifica normalizarea in cadrul proiectului nostru, am ales cazul in care entitatea PROFILE contine toate campurile ce descriu adresa (city, county, country).

Consideram tabelul urmator:

profile_id	address
11	Strada 11, Bucuresti, judetul: Bucuresti, tara: Romania
12	Strada 12, Busteni, judetul: Prahova, tara: Romania
13	Strada 13, Sinaia, judetul: Prahova, tara: Romania
14	Strada 14, Bucuresti, judetul: Bucuresti, tara: Romania
15	Strada 15, Roman, judetul: Neamt, tara: Romania
16	Strada 16, Piatra-Neamt, judetul: Neamt, tara: Romania

Aducem tabelul in forma normala 1 (FN1):

profile_id	address	city	county	country
11	Strada 11	Bucuresti	Bucuresti	Romania
12	Strada 12	Busteni	Prahova	Romania
13	Strada 13	Sinaia	Prahova	Romania
14	Strada 14	Bucuresti	Bucuresti	Romania
15	Strada 15	Roman	Neamt	Romania
16	Strada 16	Piatra-Neamt	Neamt	Romania

Observam ca e deja si in forma normala 2 (FN2), dar pentru utilizatori diferiti, avem campurile city, county, country ce se repeta. Aducem tabelul in forma FN3:

profile_id	address	city_id
11	Strada 11	1
12	Strada 12	2
13	Strada 13	3
14	Strada 14	1
15	Strada 15	4
16	Strada 16	5

city_id	name	county_id
1	Bucuresti	1
2	Busteni	2
3	Sinaia	2
4	Roman	3
5	Piatra-Neamt	3

county_id	name	country_id
1	Bucuresti	1
2	Prahova	1
3	Neamt	1

country_id	name
1	Romania

## 6. Denormalizare

Pentru a exemplifica denormalizarea in cadrul proiectului nostru, am ales cazul in care pentru un utilizator nu am mai retinut totalitatea comentariilor date. Cu alte cuvinte, avem urmatoarele tabele:

comment_picture_id	user_id	picture_id
1	1	11
2	2	12

comment_album_id	user_id	album_id
1	1	21
2	2	22

comment_status_id	user_id	status_id
1	1	31
2	2	32

Ni se cere sa aflam pentru un utilizator dat totalitatea comentariilor postate de acestea. Avand modelul exemplificat mai sus prin cele 3 tabele, ar trebui sa facem JOIN pentru fiecare dintre cele 3 tabele cu entitatea USER si apoi sa colectam rezultatul.

Solutia aceasta e aplicabila doar pentru volum mic de date, dar avand in vedere profilul aplicatiei noastre – retea de socializare – volumul de date va creste exponential.

Asadar, alegem varianta pastrarii de informatii redundante intr-un tabel suplimentar:

comment_id	user_id	comment_type_id
1	1	11
2	1	21
3	1	31
4	2	12
5	2	22
6	2	32

Astfel, pentru a vizualiza comentariile postate de un utilizator va fi suficienta interogarea tabelului COMMENT, fara a mai fi necesare JOIN-uri multiple.

## 7. Crearea bazei de date

```
CREATE TABLE USER(  
    user_id number(6),  
    language_id number(6),  
    username nvarchar2(255),  
    password nvarchar2(255),  
    first_name nvarchar2(255),  
    last_name nvarchar2(255),  
    email nvarchar2(255),  
    active nchar(1),  
    created_at date,  
    CONSTRAINT user_id_pk PRIMARY KEY (user_id),  
    CONSTRAINT lang_fk FOREIGN KEY(language_id) references LANGUAGE(language_id)  
);
```

```
CREATE TABLE PROFILE (  
    profile_id number(6),  
    user_id number(6),  
    privacy_id number(6),  
    city_id number(6),  
    date_of_birth date,  
    about_me nvarchar2(255),  
    relationship nvarchar2(255),  
    phone nvarchar2(255),  
    interests nvarchar2(255),  
    education nvarchar2(255),  
    hobbies nvarchar2(255),  
    fav_movies nvarchar2(255),  
    fav_artists nvarchar2(255),  
    fav_books nvarchar2(255),  
    address nvarchar2(255),  
    CONSTRAINT profile_id_pk PRIMARY KEY (profile_id),  
    CONSTRAINT user_profile_fk FOREIGN KEY (user_id) REFERENCES USER(user_id)  
    CONSTRAINT prof_priv_fk FOREIGN KEY (privacy_id) REFERENCES PRIVACY(privacy_id),  
    CONSTRAINT prof_city_fk FOREIGN KEY (city_id) REFERENCES CITY(city_id)  
);
```

```
CREATE TABLE CITY (  
    city_id number(6),  
    county_id number(6),
```

```

privacy_id number(6) FOREIGN KEY REFERENCES PRIVACY(privacy_id),
name nvarchar2(255),
CONSTRAINT city_id_pk PRIMARY KEY (city_id),
CONSTRAINT county_city_fk FOREIGN KEY (county_id) REFERENCES COUNTY(county_id)
);

```

```

CREATE TABLE COUNTY (
    county_id number(6),
    country_id number(6),
    privacy_id number(6) FOREIGN KEY REFERENCES PRIVACY(privacy_id),
    name nvarchar2(255),
    CONSTRAINT county_id_pk PRIMARY KEY (county_id),
    CONSTRAINT county_country_fk FOREIGN KEY (country_id) REFERENCES
COUNTY(country_id)
);

```

```

CREATE TABLE COUNTRY (
    country_id number(6),
    privacy_id number(6),
    name nvarchar2(255),
    CONSTRAINT country_id_pk PRIMARY KEY (country_id),
    CONSTRAINT prof_country_fk FOREIGN KEY (privacy_id) REFERENCES PRIVACY(privacy_id),
);

```

```

CREATE TABLE LANGUAGE (
    language_id number(6),
    name nvarchar2(255),
    CONSTRAINT language_id_pk PRIMARY KEY (language_id)
);

```

```

CREATE TABLE FRIEND_LIST(
    friend_list_id number(6) PRIMARY KEY,
    user_id number(6) FOREIGN KEY REFERENCES USER(user_id),
    friend_user_id number(6) FOREIGN KEY REFERENCES USER(user_id),
    state nchar(1),
    created_at date
);

```

```

CREATE TABLE MESSAGE(
    message_id number(6),
    user_id number(6),

```

```

friend_user_id number(6),
message_content nvarchar2(255),
created_at date,
is_read nchar(1),
is_spam nchar(1),
CONSTRAINT message_id_pk PRIMARY KEY (message_id),
CONSTRAINT mes_user_fk FOREIGN KEY (user_id) REFERENCES USER(user_id),
CONSTRAINT mes1_user_FK FOREIGN KEY (friend_user_id) REFERENCES USER(user_id)
);

```

```

CREATE TABLE STATUS (
    status_id number(6),
    user_id number(6),
    privacy_id number(6) FOREIGN KEY REFERENCES PRIVACY(privacy_id),
    status_content NVARCHAR2(255),
    created_at date,
    updated_at date,
    numberOfLikes int,
    CONSTRAINT status_pk PRIMARY KEY (status_id),
    CONSTRAINT status_user_fk FOREIGN KEY (user_id) REFERENCES USER(user_id)
);

```

```

CREATE TABLE PRIVACY(
    privacy_id number(6),
    privacy_type nvarchar2(255),
    CONSTRAINT priv_pk PRIMARY KEY (privacy_id)
);

```

```

CREATE TABLE PICTURE(
    picture_id number(6),
    album_id number(6),
    privacy_id number(6) FOREIGN KEY REFERENCES PRIVACY(privacy_id),
    description nvarchar2(255),
    created_at date,
    updated_at date,
    is_album_cover nchar(1),
    is_profile_picture nchar(1),
    numberOfLikes int,
    CONSTRAINT pic_pk PRIMARY KEY (picture_id),
    CONSTRAINT pic_album_fk FOREIGN KEY (album_id) REFERENCES ALBUM(album_id)
);

```

```

CREATE TABLE ALBUM(
  album_id number(6),
  user_id number(6),
  privacy_id number(6) FOREIGN KEY REFERENCES PRIVACY(privacy_id),
  name nvarchar2(255),
  description nvarchar2(255),
  created_at date,
  updated_at date,
  numberOfLikes int,
  CONSTRAINT album_pk PRIMARY KEY (album_id),
  CONSTRAINT album_user_fk FOREIGN KEY (user_id) REFERENCES USER(user_id)
);

```

```

CREATE TABLE COMMENT(
  comment_id number(6),
  user_id number(6) FOREIGN KEY REFERENCES USER(user_id),
  comment_type nvarchar2(32), -- Type of the comment to know in what table need to look
  comment_type_id number(6),
  CONSTRAINT comm_pk PRIMARY KEY (comment_id)
);

```

```

CREATE TABLE COMMENT_PICTURE(
  comment_picture_id number(6),
  user_id number(6),
  picture_id number(6),
  comment_content nvarchar2(255),
  created_at date,
  updated_at date,
  numberOfLikes int,
  CONSTRAINT cp_pk PRIMARY KEY (comment_picture_id),
  CONSTRAINT cP_comm_fk FOREIGN KEY (user_id) REFERENCES USER(user_id),
  constraint cp_pic_fk FOREIGN KEY (picture_id) REFERENCES PICTURE(picture_id)
);

```

```

CREATE TABLE COMMENT_ALBUM(
  comment_album_id number(6),
  user_id number(6),
  album_id number(6),
  comment_content nvarchar2(255),
  created_at date,
  updated_at date,

```



```

    numberOfLikes int,
    CONSTRAINT ca_pk PRIMARY KEY (comment_album_id),
    CONSTRAINT ca_comUSR_fk FOREIGN KEY (user_id) REFERENCES USER(user_id),
    CONSTRAINT ca_album_fk FOREIGN KEY (album_id) REFERENCES ALBUM(album_id)
);

```

```

CREATE TABLE COMMENT_STATUS(
    comment_status_id number(6),
    user_id number(6),
    status_id number(6),
    comment_content nvarchar2(255),
    created_at date,
    updated_at date,
    numberOfLikes int,
    CONSTRAINT cs_pk PRIMARY KEY (comment_status_id),
    CONSTRAINT cs_comUSR_fk FOREIGN KEY (user_id) REFERENCES USER(user_id),
    CONSTRAINT status_cs_fk FOREIGN KEY (status_id) REFERENCES STATUS(status_id)
);

```

```

CREATE TABLE LIKE(
    like_id number(6) CONSTRAINT l_nn NOT NULL,
    user_id number(6),
    like_type_id number(6),
    like_type nvarchar2(32),
    CONSTRAINT l_pk PRIMARY KEY (like_id),
    CONSTRAINT l_fr_fk FOREIGN KEY (user_id) REFERENCES USER(user_id)
);

```

```

CREATE TABLE LIKE_PICTURE(
    like_picture_id number(6),
    user_id number(6),
    picture_id number(6),
    created_at date,
    CONSTRAINT lp_pk PRIMARY KEY (like_picture_id),
    CONSTRAINT l_lp_fk FOREIGN KEY (user_id) REFERENCES USER(user_id),
    CONSTRAINT lp_p_fk FOREIGN KEY (picture_id) REFERENCES PICTURE(picture_id)
);

```

```

CREATE TABLE LIKE_ALBUM(
    like_album_id number(6),
    user_id number(6),

```

```

album_id number(6),
created_at int,
CONSTRAINT la_pk PRIMARY KEY (like_album_id),
CONSTRAINT l_la_fk FOREIGN KEY (user_id) REFERENCES USER(user_id),
CONSTRAINT la_a_fk FOREIGN KEY (album_id) REFERENCES ALBUM(album_id)
);

```

```

CREATE TABLE LIKE_STATUS(
like_status_id number(6),
user_id number(6),
status_id number(6),
created_at int,
CONSTRAINT ls_pk PRIMARY KEY (like_status_id),
CONSTRAINT ls_l_fk FOREIGN KEY (user_id) REFERENCES USER(user_id),
CONSTRAINT st_l_fk FOREIGN KEY (status_id) REFERENCES STATUS(status_id)
);

```

```

CREATE TABLE LIKE_COMMENT(
like_comment_id number(6),
user_id number(6),
comment_id number(6),
created_at int,
CONSTRAINT lk_pk PRIMARY KEY (like_comment_id),
CONSTRAINT l_lk_fk FOREIGN KEY (user_id) REFERENCES USER(user_id),
CONSTRAINT c_lk_fk FOREIGN KEY (comment_id) REFERENCES COMMENT(comment_id)
);

```

```

CREATE TABLE EVENT(
event_id number(6),
user_id number(6) FOREIGN KEY REFERENCES USER(user_id),
city_id number(6) FOREIGN KEY REFERENCES CITY(city_id),
name nvarchar2(255),
description nvarchar2(255),
address number(6),
date date,
created_at date,
updated_at date,
CONSTRAINT e_pk PRIMARY KEY(event_id),
);

```

```

CREATE TABLE PARTICIPATE_EVENT(

```

```

participate_event_id number(6),
event_id number(6),
user_id number(6),
CONSTRAINT ev_p_pk PRIMARY KEY (p_event_id),
CONSTRAINT ev_p_fk FOREIGN KEY (event_id) REFERENCES EVENT(event_id),
CONSTRAINT us_p_fk FOREIGN KEY (user_id) REFERENCES USER(user_id)
);

```

```

CREATE TABLE FEED(
    feed_id number(6),
    user_id number(6),
    feed_type_id number(6),
    CONSTRAINT feed_pk PRIMARY KEY (feed_id),
    CONSTRAINT feed_u_fk FOREIGN KEY (user_id) REFERENCES USER(user_id)
);

```

```

CREATE TABLE FEED_PICTURE(
    feed_picture_id number(6),
    picture_id number(6),
    user_id number(6),
    CONSTRAINT feed_p_pk PRIMARY KEY (feed_picture_id),
    CONSTRAINT fp_pic_fk FOREIGN KEY (picture_id) REFERENCES PICTURE(picture_id),
    CONSTRAINT fp_feed_fk FOREIGN KEY (user_id) REFERENCES USER(user_id)
);

```

```

CREATE TABLE FEED_ALBUM(
    feed_album_id number(6),
    album_id number(6),
    user_id number(6),
    CONSTRAINT feed_a_pk PRIMARY KEY (feed_album_id),
    CONSTRAINT fa_al_fk FOREIGN KEY (album_id) REFERENCES ALBUM(album_id),
    CONSTRAINT fp_feed_fk FOREIGN KEY (user_id) REFERENCES USER(user_id)
);

```

```

CREATE TABLE FEED_STATUS(
    feed_status_id number(6),
    status_id number(6),
    user_id number(6),
    CONSTRAINT feed_s_pk PRIMARY KEY (feed_status_id),
    CONSTRAINT fs_usr_fk FOREIGN KEY (user_id) REFERENCES USER(user_id),
    CONSTRAINT fs_st_fk FOREIGN KEY (status_id) REFERENCES STATUS(status_id),

```

);

```
CREATE TABLE FEED_EVENT(  
    feed_event_id number(6),  
    event_id number(6),  
    user_id number(6),  
    CONSTRAINT feed_e_pk PRIMARY KEY (feed_event_id),  
    CONSTRAINT fe_usr_fk FOREIGN KEY (user_id) REFERENCES USER(user_id),  
    CONSTRAINT fe_ev_fk FOREIGN KEY (event_id) REFERENCES EVENT(event_id)  
);
```

```
CREATE TABLE NOTIFICATION(  
    notification_id number(6) PRIMARY KEY,  
    user_id number(6) FOREIGN KEY REFERENCES USER(user_id),  
    notification_type_id number(6),  
    notification_type nvarchar2(32)  
)
```

```
CREATE TABLE NOTIFICATION_COMMENT(  
    notification_comment_id number(6),  
    comment_id number(6),  
    user_id number(6) FOREIGN KEY REFERENCES USER(user_id),  
    content nvarchar2(255),  
    CONSTRAINT not_comm_pk PRIMARY KEY (notification_comment_id),  
    CONSTRAINT com_not_fk FOREIGN KEY (comment_id) REFERENCES COMMENT(comment_id)  
);
```

```
CREATE TABLE NOTIFICATION_LIKE(  
    notification_like_id number(6),  
    like_id number(6),  
    user_id number(6) FOREIGN KEY REFERENCES USER(user_id),  
    content nvarchar2(255),  
    CONSTRAINT not_lk_pk PRIMARY KEY (notification_like_id),  
    CONSTRAINT lik_not_fk FOREIGN KEY (like_id) REFERENCES LIKE(like_id)  
);
```

```
CREATE TABLE NOTIFICATION_FRIEND(  
    notification_firend_id number(6),  
    friend_id number(6) FOREIGN KEY REFERENCES USER(user_id),  
    user_id number(6) FOREIGN KEY REFERENCES USER(user_id),  
    content nvarchar2(255),
```

```
CONSTRAINT not_fr_pk PRIMARY KEY (notification_friend_id)
);
```

## 8. Ce se doreste de la model

Modelul nostru isi propune sa implementeze functionalitatile de baza ale unei retele de socializare. Dintre acestea amintim :

- Posibilitatea crearii unui cont de utilizator si completarea unui profil
- Posibilitatea editarii tuturor informatiilor prezente in cadrul profilului
- Vizualizarea profilelor altor utilizatori
- Postarea, editarea si stergerea de statusuri, poze, albume
- Publicarea, editarea, stergerea comentariilor si/ sau aprecierea anumitor continuturi apartinand altor utilizatori din lista de prieteni
- Trimiterea cererilor de prietenie catre alti utilizatori, urmata sau nu de aprobarea acestora
- Comunicarea, prin intermediul mesajelor, cu utilizatorii din lista de prieteni
- Crearea si participarea la evenimente
- Notificarea utilizatorului asupra interactiunii altor utilizatori cu continutul postat de el
- Accesarea ultimelor noutati in ceea ce ii priveste pe utilizatorii din lista de prieteni

## 9. Diagrama conceptuala a modelului

