

Heuleum

Miroljub Mihailovic 1163471

Giacomo Miola 1165238

Giovanni Pinna 1146903

a.a. 2019/2020

Indice

1 Descrizione del progetto.....	3
1.1 Analisi dei requisiti	3
1.2 Operazioni sulla base di dati	4
1.3 Glossario dei termini.....	5
1.4 Strutturazione dei requisiti	6
2 Progettazione Concettuale.....	7
2.1 Modello Concettuale: Entita`-Associazione (E-R)	7
2.2 Dizionario dei dati	8
2.2.1 Entita`	8
2.2.2 Relazioni	8
2.3 Schema Concettuale, Regole di vincolo	9
3 Progettazione Logica.....	10
3.1 Ristrutturazione schema E.R.	10
3.2 Modello Logico: Relazionale	11
3.3 Vincolo di integrità referenziale	12
4 Codice SQL	13
4.1 Struttura.....	13
4.2 Query.....	18

1. Descrizione del progetto

Nell'ultimo periodo ha preso piede la tendenza di condividere in tempo reale la propria vita, passioni e hobby. Per far fronte a tale necessità sono nati vari servizi di live streaming, tra questi, Heuleum, un servizio che racchiude le migliori funzionalità, ovvero live streaming, social network e video on demand. All'interno della piattaforma è possibile interagire con utenti sulla base di interessi e passioni comuni ed avere la possibilità di trarre un guadagno dai propri contenuti condivisi.

1.1 Analisi dei requisiti

Si vuole realizzare una base di dati per gestire la piattaforma, organizzando le informazioni degli utenti e dei vari contenuti condivisi. Per accedere ai servizi, l'utente dovrà registrarsi sulla piattaforma fornendo un indirizzo email, password e un username pubblico. Se l'utente lo desidera può personalizzare il proprio profilo aggiungendo una biografia ed eventualmente impostare un'immagine profilo, viene inoltre registrato lo stato dell'utente (online, offline), il numero totale di visualizzazioni del profilo (views) e di seguaci (followers). Una volta che un utente viene inserito nella base di dati, gli verrà assegnato automaticamente un identificatore univoco. Dopo la registrazione l'utente può:

1. Avviare una live stream.
2. Guardare una live stream o un video.
3. Seguire un utente oppure seguire una categoria in base ai propri interessi.
4. Supportare (subscribe) un altro utente mediante un'iscrizione a pagamento con rinnovo mensile, scegliendo tra 3 importi predefiniti (5€, 10€, 15€).
5. Inviare o accettare richieste di amicizia.
6. Pubblicare una live stream conclusa come un video.
7. Creare un evento per avvisare gli utenti di un'importante live stream fuori programma, specificando il nome, la descrizione, la categoria, la data di inizio e di fine.

Durante la visione di una live stream un utente può:

1. Partecipare ad una chat di gruppo con tutti gli spettatori, dove ogni messaggio inviato è caratterizzato dal contenuto testuale e dalla data di invio.
2. Creare pubblicamente un video di breve durata, ovvero una clip, registrando un intervallo della live stream. Ogni clip viene rappresentata dal momento di inizio e fine rispetto alla live stream, inoltre viene registrata la data di creazione e il numero totale di visualizzazioni (views).
3. Prendere parte a un team, ovvero una squadra di giocatori gareggianti in uno stesso videogioco, alla quale viene associato un nome e una descrizione.

Ogni live stream è contraddistinta dall'identificativo del file multimediale associato, un titolo, la categoria di contenuti, il momento di inizio, il numero totale di spettatori correnti (viewers), la regola di partecipazione alla chat, la quale riguarda solo il permesso di inviare messaggi. Ci sono tre regole selezionabili: tutti possono scrivere, solo i followers possono scrivere, solo i subscribers possono scrivere.

Il video, similmente alla live, memorizza l'identificativo del file multimediale associato, un titolo e la categoria di contenuti, inoltre memorizza il momento temporale di fine della live, il numero totale di visualizzazioni e mantiene validi i messaggi e clip relative alla corrispondente live. L'utente, qualora decidesse di rendere il video pubblico, quest'ultimo potrà essere visionato interamente on demand dagli utenti. In caso contrario, solo le clip registrate relative ad esso saranno visibili.

Una categoria rappresenta un determinato ambito di contenuti ed è contraddistinta da un nome e dal numero totale di spettatori (viewers), ovvero dalla somma degli spettatori correnti di tutte le live stream relative alla suddetta categoria.

Viene data la possibilità di associare ad ogni categoria e live stream uno o più tag, ovvero una breve parola che sintetizza una caratteristica, per facilitare la ricerca di contenuti specifici.

1.2 Operazioni sulla base di dati

Operazione 1: avviare una live stream;

Operazione 2: concludere una live ed eventualmente pubblicarla come video;

Operazione 3: registrare una clip durante la visione di una live;

Operazione 4: mandare, se permesso, un messaggio nella chat di una live stream;

Operazione 5: visualizzare messaggi nella chat di una live stream;

Operazione 6: guardare (o smettere di guardare) una live stream;

Operazione 7: creare (o eliminare) un evento;

Operazione 8: partecipare (o togliersi) ad un team;

Operazione 9: seguire (o smettere di seguire) altri utenti;

Operazione 10: supportare un utente mediante un'iscrizione;

Operazione 11: mandare richieste di amicizia ad altri utenti;

Operazione 12: accettare o rifiutare la richiesta di amicizia ricevuta;

Operazione 13: seguire (o smettere di seguire) una o più categorie;

Operazione 14: visualizzare il numero di spettatori di una live stream;

Operazione 15: visualizzare il numero totale di spettatori di tutte live stream per una data categoria;

Operazione 16: visualizzare il numero di followers di un utente;

Operazione 17: visualizzare la lista delle live degli utenti seguiti;

Operazione 18: visualizzare le live che stanno guardando gli amici di un utente;

Operazione 19: ricercare live stream o video pubblicati che contengono un certo tag o una certa parola nel titolo;

Operazione 20: ricercare un utente per nome;

Operazione 21: visualizzare le top 10 live stream con il maggior numero di spettatori, relative agli streamer seguiti dall'utente;

Operazione 22: visualizzare le live con il maggior numero di spettatori per ogni categoria.

1.3 Glossario dei termini

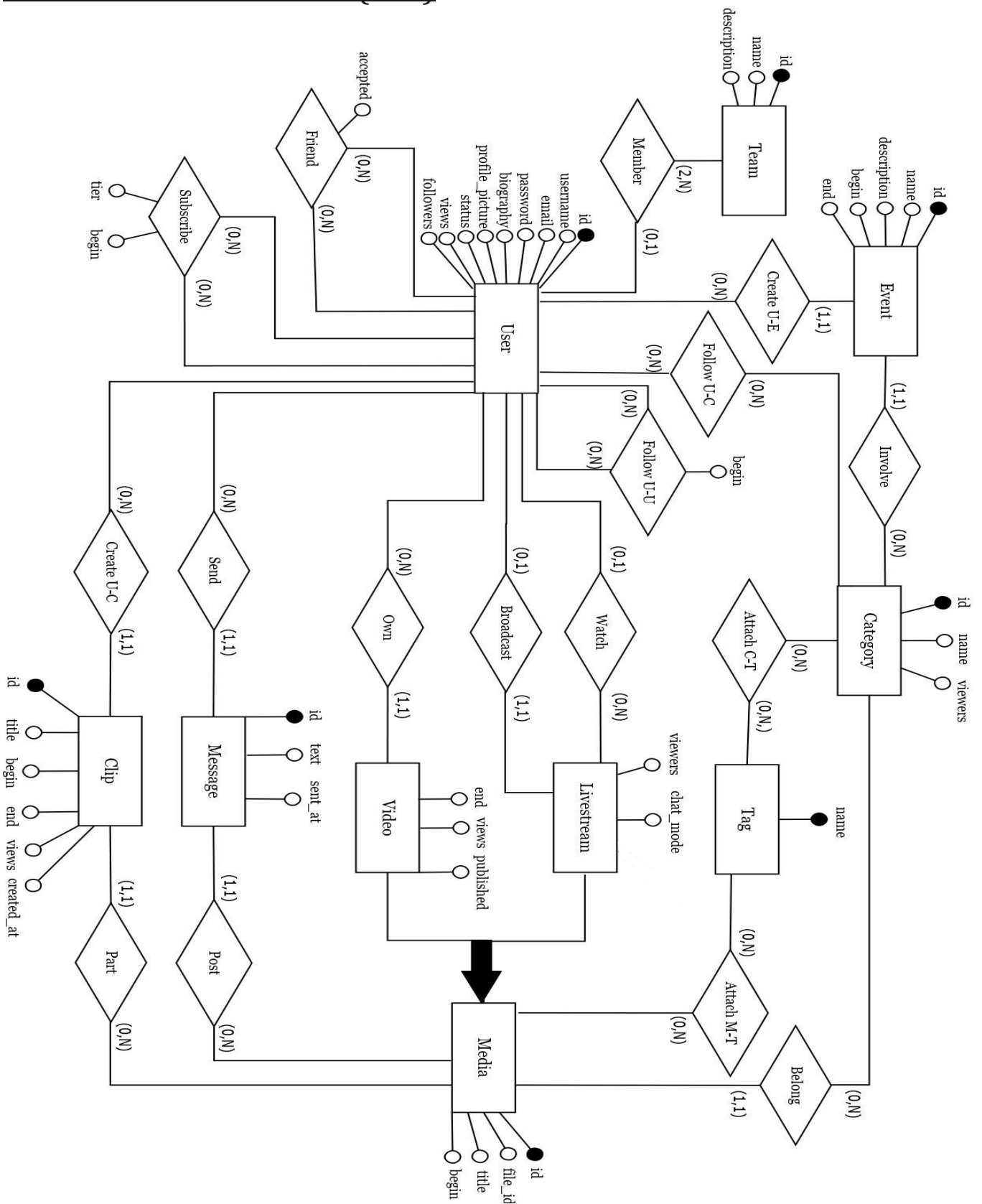
Termine	Descrizione	Sinonimi	Collegamenti
Utente	Colui che usufruisce dei servizi	User	Live stream, Video, Categoria, Team, Utente, Messaggio, Video
Live stream	Un video trasmesso in tempo reale da un Utente	Live/Stream	Utente, Categoria, Tag, Clip, Messaggio
Video	Live stream terminata, che può essere pubblicata per essere visualizzata in ogni momento		Utente, Categoria Tag, Clip, Messaggio
Categoria	Ambito di contenuti	Category	Utente, Live stream, Video, Tag, Evento
Tag	Breve parola che sintetizza una caratteristica		Live stream, Video Categoria
Messaggio	Contenuto testuale inviato da un Utente in una chat	Message	Utente, Live stream Video
Clip	Frammento di una Live stream creato da uno spettatore		Utente, Live stream, Video
Evento	Live stream programmata per una certa data, creata da un Utente	Event	Utente, Categoria
Squadra	Gruppo di almeno due Utenti che gareggiano allo stesso gioco	Team	Utente

1.4 Strutturazione dei requisiti

Frase relative ad Utente
Per accedere ai servizi, l'utente dovrà registrarsi sulla piattaforma fornendo un indirizzo email, password e un'username pubblico. Se l'utente lo desidera può personalizzare il proprio profilo aggiungendo una biografia ed eventualmente impostare un'immagine profilo. Inoltre, viene registrato lo stato dell'utente (online, offline, invisibile), il numero totale di visualizzazioni del profilo (views) e di seguaci (followers). Una volta che un utente viene inserito nella base di dati, gli verrà assegnato automaticamente un identificatore univoco. Viene inoltre data la possibilità di eliminare il proprio account dalla piattaforma. Ciò comporta una rimozione di tutte le informazioni, contenuti e attività relative all'utente.
Frase relative a Live stream
Ogni live stream è contraddistinta dall'identificativo del file multimediale associato, un titolo, la categoria di contenuti, il momento di inizio, il numero totale di spettatori correnti (viewers), la regola di partecipazione alla chat, la quale riguarda solo il permesso di inviare messaggi. Ci sono tre regole selezionabili: tutti possono scrivere, solo i followers possono scrivere, solo i subscribers possono scrivere.
Frase relative a Video
Il video, similmente alla live, memorizza l'identificativo del file multimediale associato, un titolo e la categoria di contenuti, inoltre memorizza il momento temporale di fine della live, il numero totale di visualizzazioni e mantiene validi i messaggi e clip relative alla corrispondente live. L'utente, qualora decidesse di rendere il video pubblico, quest'ultimo potrà essere visionato interamente on demand dagli utenti. In caso contrario, solo le clip registrate relative ad esso saranno visibili.
Frase relative a Messaggio
Ogni messaggio inviato è caratterizzato dal contenuto testuale e dalla data di invio.
Frase relative a Clip
Video di breve durata, che registra un intervallo della live stream, per mettere in evidenza un momento significativo. Ogni clip viene rappresentata dal momento di inizio e fine rispetto alla livestream, inoltre viene registrata la data di creazione e il numero totale di visualizzazioni (views).
Frase relative a Categoria
Una categoria rappresenta un determinato ambito di contenuti ed è contraddistinta da un nome e dal numero totale di spettatori (viewers), ovvero dalla somma degli spettatori correnti di tutte le livestream relative alla suddetta categoria.
Frase relative a Tag
Breve parola che sintetizza una caratteristica, per facilitare la ricerca di contenuti specifici.
Frase relative ad Evento
Modo per avvisare gli utenti di un'importante live stream fuori programma, specificandone il nome, la descrizione, la categoria di appartenenza, la data di inizio e di fine.
Frase relative a Team
Una squadra di giocatori gareggianti in uno stesso videogioco, alla quale viene associato un nome e una descrizione.

2. Progettazione Concettuale

2.1 Entita'-Associazione (E-R)



2.2 Dizionario dei dati

Entità

Entita`	Descrizione	Attributi	Identificatore
User	Utilizzatore della piattaforma	id, username, email, password, biography, profile picture, status, views, followers	id
Media	Un generico file multimediale	id, file id, title, begin	id
Live Stream	Un video trasmesso in diretta live	viewers, chat mode, chat mode age	id
Video	Una Live stream terminata	end, views, published	id
Message	Un messaggio testuale inviato nella chat di una Live stream	id, text, sent at	id
Clip	Un frammento di una Live stream creato da uno spettatore	id, title, begin, end, views, created at	id
Categoria	Ambito di contenuti	id, name	id
Tag	Parola che sintetizza una caratteristica di un Media	name	name
Event	Live stream programmata organizzata da un User	id, name, description, begin, end	id
Team	Gruppo uno o piu utenti` che gareggiano allo stesso gioco	id, name, description	id

Relazioni

Relazione	Descrizione	Cardinalita`	Attributi
Watch	Associa un User alla Live stream che sta guardando	User(0, 1) Livestream(0, N)	
Broadcast	Associa un User alla Live stream che sta trasmettendo	User(0, 1) Livestream(1, 1)	
Own	Associa un User ad un Video	User(0, N) Video(1, 1)	
Belong	Associa i Media alla Category di appartenenza	Media(1, 1) Category(0, N)	
Attach M-T	Associa il Media ai Tag di appartenenza	Media(0, N) Tag(0, N)	
Attach C-T	Associa la Category ai Tag di appartenenza	Category(0, N) Tag(0, N)	
Follow U-U	Associa un User follower all'User seguito (streamer)	User(0, N) User(0, N)	begin
Follow U-C	Associa un User alle Category di suo interesse	User(0, N) Category(0, N)	

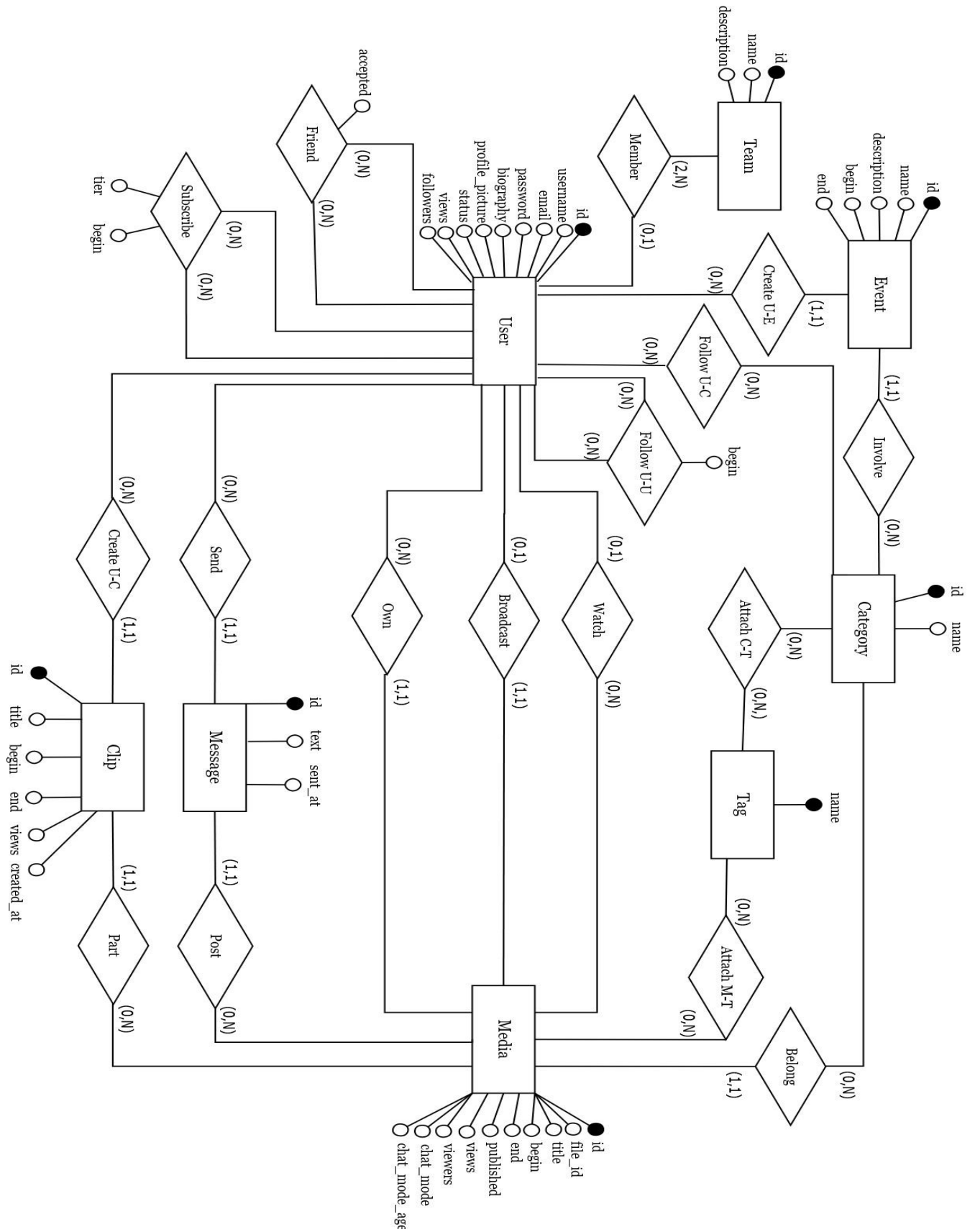
Friend	Associa due Users con una relazione di amicizia se la richiesta viene accettata	User(0, N) User(0, N)	accepted
Subscribe	Associa un User subscriber all'utente che vuole supportare (streamer)	User(0, N) User(0, N)	tier, begin
Member	Associa gli Users al proprio Team di appartenenza	User(0, 1) Team(2, N)	
Send	Associa un User ai propri messaggi inviati	User(0, N) Message(1, 1)	
Post	Associa i Messages inviati nella chat relativa alla Media	Live stream(1, 1) Message(0, N)	
Create U-C	Associa un User alle proprie Clip create	User(0, N) Clip(1, 1)	
Part	Associa i Clip creati alla relativa Media	Media(1, 1) Clip(0, N)	
Create U-E	Associa un User al proprio Event creato	User(0, N) Event(1, 1)	

2.3 Schema Concettuale, Regole di vincolo

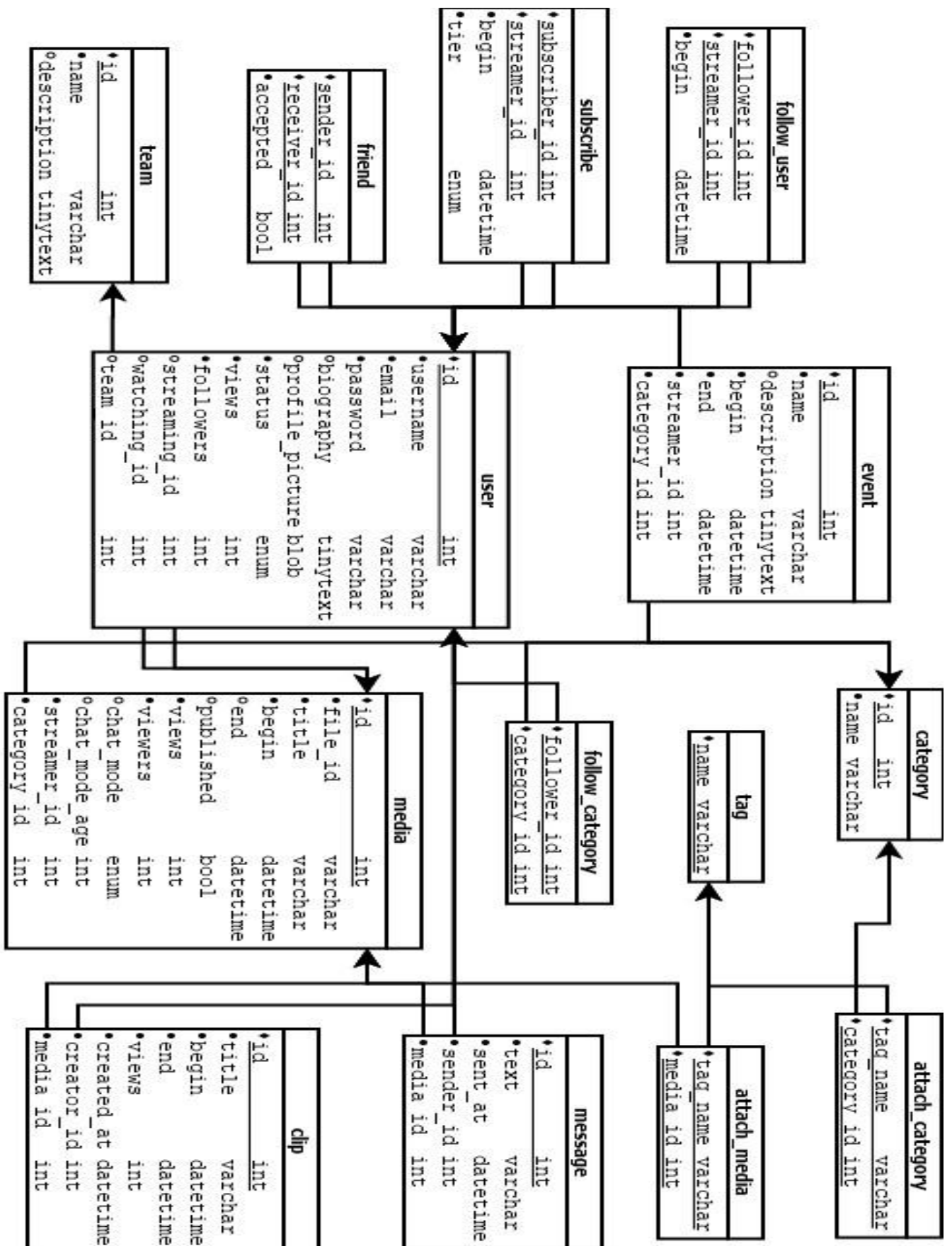
- L'inizio di una clip deve essere successivo all'inizio della live stream o del video a cui si riferisce;
- La fine di una clip, di un video e di un evento devono essere successive al corrispondente inizio, inoltre, la fine di una clip deve essere inferiore al momento corrente della live stream o alla fine del video, secondo a quale dei due si riferisce;
- Il numero totale di spettatori correnti di una live stream e di una categoria, di visualizzazioni e di followers di un profilo utente e di visualizzazioni un video devono essere maggiori o uguali a zero;
- Un utente non può seguire, iscriversi o mandare richieste di amicizia a se stesso;
- Per potere mandare un messaggio nella chat di una live o per creare una clip relativa ad essa, l'utente deve essere spettatore;
- Non possono esistere due categorie o due team con lo stesso nome;
- Il contatore viewers di una live stream deve essere aggiornato quando un utente inizia o finisce di guardarla;
- Il contatore viewers di una categoria deve essere aggiornato quando un utente inizia o finisce di guardare una live relativa a tale categoria;
- Il contatore followers di un utente deve essere aggiornato quando quest'ultimo acquisisce o perde un seguace;
- Il contatore views di un utente deve essere aggiornato quando il suo profilo viene visualizzato da un altro utente;
- Il contatore views di un video deve essere aggiornato quando quest'ultimo viene visualizzato da un altro utente.

3 Progettazione Logica

3.1 Ristrutturazione schema E.R.



3.2 Modello Logico: Relazionale



user(id, username, email, password, biography*, profile picture*, status, views, followers, streaming id*, watching id*, team id*)

media(id, file id, title, begin, end*, published*, views, viewers, chat mode*, chat mode age*, streamer id, category id)

category(id, name)

event(id, name, description*, begin, end, streamer id, category id)

tag(name)

team(id, name, description*)

follow_user(follower id, streamer id, begin)

follow_category(follower id, category id)

friend(sender id, receiver id, accepted)

subscribe(subscriber id, streamer id, begin, tier)

attach_media(tag name, media id)

attach_category(tag name, category id)

message(id, text, sent at, sender id, media id)

clip(id, title, begin, end, views, created at, creator id, media id)

3.3 Vincoli di integrità referenziale

Relazione	Vincoli	
user	streaming id → media.id watching id → media.id team id → team.id	
media	category id → category.id	
event	streamer id → user.id	category id → category.id
follow user	follower .id → user.id	streamer id → user.id
follow category	follower .id → user.id	category id → category.id
friend	sender id → user.id	receiver id → user.id
subscribe	subscriber id → user.id	streamer id → user.id
attach media	tag name → tag.name	media id → media.id
attach category	tag name → tag.name	category id → category.id
message	sender id → user.id	media id → media.id
clip	creator id → user.id	media id → media.id

4 Codice SQL

4.1 Struttura

create table user_

```
(  
id serial primary key,  
username varchar(20) not null,  
email varchar(50) not null,  
password varchar(30) not null,  
biography varchar(300),  
profile_picture varchar(200),  
status varchar(11) default 'offline',  
views integer not null check(views>=0),  
followers integer not null check(followers>=0),  
streaming_id integer,  
watching_id integer,  
team_id integer,  
foreign key (streaming_id) references media(id),  
foreign key (watching_id) references media(id),  
foreign key (team_id) references team(id)  
);
```

create table media

```
(  
id serial primary key,  
file_id varchar(200) not null,  
title varchar(30) not null,  
begin timestamp not null,  
end_timestamp,  
published boolean default false,
```

```
views integer not null check(views>=0),
viewers integer not null check(viewers>=0),
chat_mode varchar(15) default 'all',
chat_mode_age integer,
streamer_id integer,
category_id integer,
foreign key (category_id) references category(id),
foreign key (streamer_id) references user_(id)
);
```

```
create table follow_user(
follower_id integer not null,
streamer_id integer not null,
begin timestamp not null,
primary key(follower_id,streamer_id),
foreign key (follower_id) references user_(id),
foreign key (streamer_id) references user_(id)
);
```

```
create table subscribe
(
subscriber_id integer not null,
streamer_id integer not null,
begin timestamp not null,
tier integer not null,
primary key(subscriber_id,streamer_id),
foreign key (subscriber_id) references user_(id),
foreign key (streamer_id) references user_(id)
);
```

create table friend

```
(  
sender_id integer not null,  
receiver_id integer not null,  
accepted boolean default false,  
primary key(sender_id,receiver_id),  
foreign key (sender_id) references user_(id),  
foreign key (receiver_id) references user_(id)  
);
```

create table event

```
(  
id serial primary key,  
name varchar(30) not null,  
description varchar(200),  
begin timestamp not null,  
end_timestamp,  
streamer_id integer,  
category_id integer,  
foreign key (streamer_id) references user_(id),  
foreign key (category_id) references category(id)  
);
```

create table follow_category

```
(  
follower_id integer not null,  
category_id integer not null,  
primary key(follower_id, category_id),  
foreign key(follower_id) references user_(id),  
foreign key(category_id) references category(id)
```

);

create table attach_category

```
(  
tag_name varchar(30) not null,  
category_id integer not null,  
primary key(tag_name, category_id),  
foreign key(tag_name) references tag(nome),  
foreign key(category_id) references category(id)  
);
```

create table attach_media

```
(  
tag_name varchar(30) not null,  
media integer not null,  
primary key(tag_name, media),  
foreign key(tag_name) references tag(nome),  
foreign key(media) references media(id)  
);
```

create table message

```
(  
id serial primary key,  
text varchar(300) not null,  
sent_at timestamp not null,  
sender_id integer not null,  
media_id integer not null,  
foreign key(sender_id) references user_(id),  
foreign key(media_id) references media(id)  
);
```


create table clip

```
(  
id serial primary key,  
title varchar(30) not null,  
begin_timestamp not null,  
end_timestamp not null,  
views integer not null check(views>=0),  
created_at timestamp not null,  
creator_id integer not null,  
media_id integer not null,  
foreign key(creator_id) references user_(id),  
foreign key(media_id) references media(id)  
);
```

create table category

```
(  
id serial primary key,  
name varchar(20) not null  
);
```

create table tag

```
(  
nome varchar(30) primary key  
);
```

create table team

```
(  
id serial primary key,  
name varchar(20) not null,  
description varchar(200)  
);
```

4.2 Query

- Titolo del video pubblicato con maggiore views

create view maxmedia as

```
(  
    select max(views) as maxm  
    from media  
    where end_ is not null  
    and published = true  
);  
select title, views  
from media  
where end_ is not null  
and published = true  
and views = ( select maxm from maxmedia);
```

- Utente/i con numero maggiore di followers

create view maxfollowers as

```
(  
    select max(followers) as maxf
```

```

        from user_
    );

select username, followers
from user_
where followers = (select maxf
                   from maxfollowers);

```

- Categoria con maggiori tag usati

```

create view countCategory as
(
select category_id, count(*) as numCategory
from attach_category
group by category_id
order by category_id
);

select c.name, cc.numcategory
from countCategory as cc join category as c on cc.category_id = c.id
where cc.numCategory = ( select max(numCategory)
                        from countCategory );

```

- Numero di componenti per ciasun team

```
select t.name,count(*) as num_team
from user_ join team as t on user_.team_id = t.id
where team_id is not null
group by t.name
;
```

- Guadagno totale per ciascun utente (id e username) che riceve donazioni

```
select streamer_id, sum(tier) as tier_totale_€
from subscribe
group by streamer_id
order by streamer_id
;
```

- Username e messaggi degli utenti che rispondono ai media che sono in live stream

```
select u.id as id_user , u.username, me.id as id_media, me.title, m.text
from (message as m join user_ as u on m.sender_id=u.id)
      join media as me on m.media_id = me.id
where me.end_ is null ;
```

- Utente (username e id) che ha creato più clip

```
create view tot_clip as
```

```
(  
    select creator_id, count(*) as num_clip  
    from clip  
    group by creator_id  
);
```

```
select u.id, u.username, c.num_clip  
from tot_clip as c join user_ as u on c.creator_id = u.id  
where c.num_clip = (select max(num_clip)  
                    from tot_clip  
                    )  
;
```

