

OmegaStream

Marco Siragna 1161572
Nicholas Miazzo 1131692

a.a. 2018/2019

Abstract

Nell'ultimo decennio ha preso piede la tendenza di condividere in tempo reale la propria vita, passioni e hobby con un vasto pubblico senza limiti fisici. Per far fronte a tale necessità sono nati vari servizi di live streaming. Tra questi si distingue OmegaStream (abbreviamo OS), un servizio unico nel suo genere che racchiude le migliori funzionalità, ovvero live streaming, social network e video on demand.

All'interno della piattaforma OS è possibile interagire con utenti sulla base di interessi e passioni comuni ed avere la possibilità di trarre un guadagno dai propri contenuti condivisi.

Nel corso degli anni, noi di OmegaStream, ci siamo sempre impegnati ad offrire un servizio efficiente e all'avanguardia. Tuttavia, avendo registrato un aumento esponenziale del numero di utenti abbiamo avuto l'esigenza di ampliare la nostra infrastruttura e allo stesso tempo migliorare i nostri servizi.

1 Analisi dei requisiti

Si vuole realizzare una base di dati per gestire la piattaforma di streaming OS, 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. 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.

Dopo la registrazione l'utente può:

- avviare una livestream;
- guardare una livestream o un video;
- seguire (follow) un utente, tenendone traccia della data, o seguire una categoria in base ai propri interessi;
- supportare (subscribe) un altro utente mediante un'iscrizione a pagamento con rinnovo mensile, scegliendo tra 3 importi predefiniti, detti tier (\$4.99, \$9.99, \$24.99), tenendo traccia della data del pagamento;
- inviare o accettare richieste di amicizia;
- pubblicare una livestream conclusa come un video;
- creare un **evento** per avvisare gli utenti di un'importante livestream fuori programma, specificandone il nome, la descrizione, la categoria di appartenenza, la data di inizio e di fine.

Durante la visione di una livestream un utente può:

- partecipare ad una chat testuale di gruppo con tutti gli spettatori, dove ogni **messaggio** inviato è caratterizzato dal contenuto testuale e dalla data di invio;
- creare pubblicamente un video di breve durata, ovvero una **clip**, registrando un intervallo della livestream, 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);
- far parte di un **team**, ovvero una squadra di giocatori gareggianti in uno stesso videogioco, alla quale viene associato un nome e una descrizione.

Ogni **livestream** 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. Inoltre, per le ultime due regole, è possibile specificare da quanto tempo (in secondi) lo spettatore deve essere follower o subscriber per poter 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 ondemand 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 livestream relative alla suddetta categoria.

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

1.1 Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Utente	Colui che usufruisce dei servizi offerti da OS	User	Livestream, Video, Categoria, Team, Utente, Messaggio, Video
Livestream	Un video trasmesso in tempo reale da un Utente	Live	Utente, Categoria, Tag, Clip, Messaggio
Video	Livestream terminata, che può essere pubblicata per essere visualizzata in ogni momento		Utente, Categoria, Tag, Clip, Messaggio
Categoria	Ambito di contenuti	Category	Utente, Livestream, Video, Tag, Evento
Tag	Breve parola che sintetizza una caratteristica		Livestream, Video, Categoria
Messaggio	Contenuto testuale inviato da un Utente in una chat	Message	Utente, Livestream, Video
Clip	Frammento di una Livestream creato da uno spettatore		Utente, Livestream, Video
Evento	Livestream 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

1.2 Operazioni

Operazione 1: avviare una livestream;

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 livestream;

Operazione 5: visualizzare messaggi nella chat di una livestream;

Operazione 6: guardare (o smettere di guardare) una livestream;

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 livestream;

Operazione 15: visualizzare il numero totale di spettatori di tutte livestream 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 livestream 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 livestream 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 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 Livestream

Ogni livestream 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. Inoltre, per le ultime due regole, è possibile specificare da quanto tempo (in secondi) lo spettatore deve essere follower o subscriber per poter 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 ondemand 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 livestream, 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

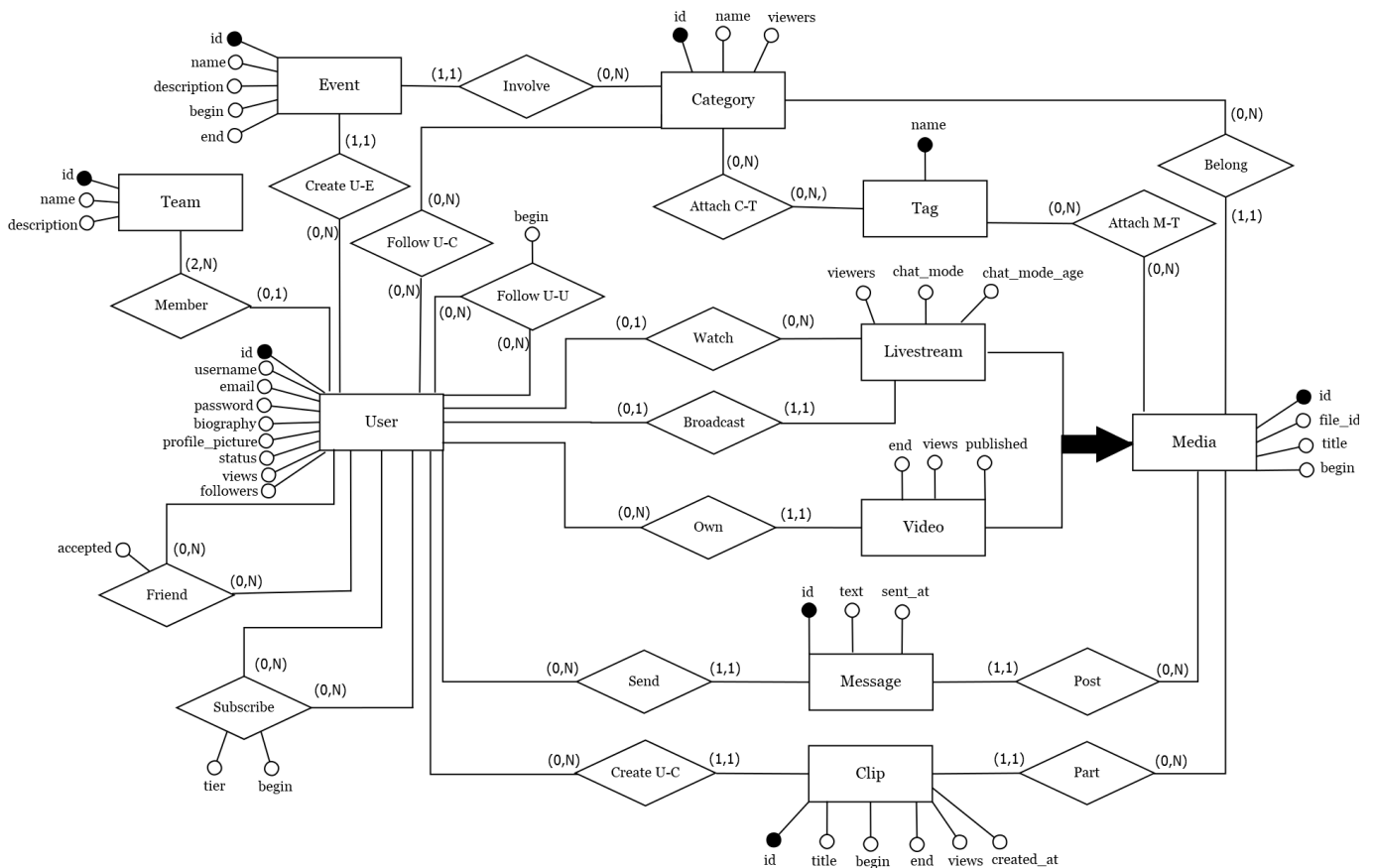
Modo per avvisare gli utenti di un'importante livestream fuori programma, specificandone il nome, la descrizione, la categoria di appartenenza, la data di inizio e di fine.

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 Diagramma E-R



2.2 Dizionario dei dati

2.2.1 Regole di vincolo

- L'inizio di una clip deve essere successivo all'inizio della livestream 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 livestream o alla fine del video, secondo a quale dei due si riferisce;
- il numero totale di spettatori correnti di una livestream 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 livestream 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.

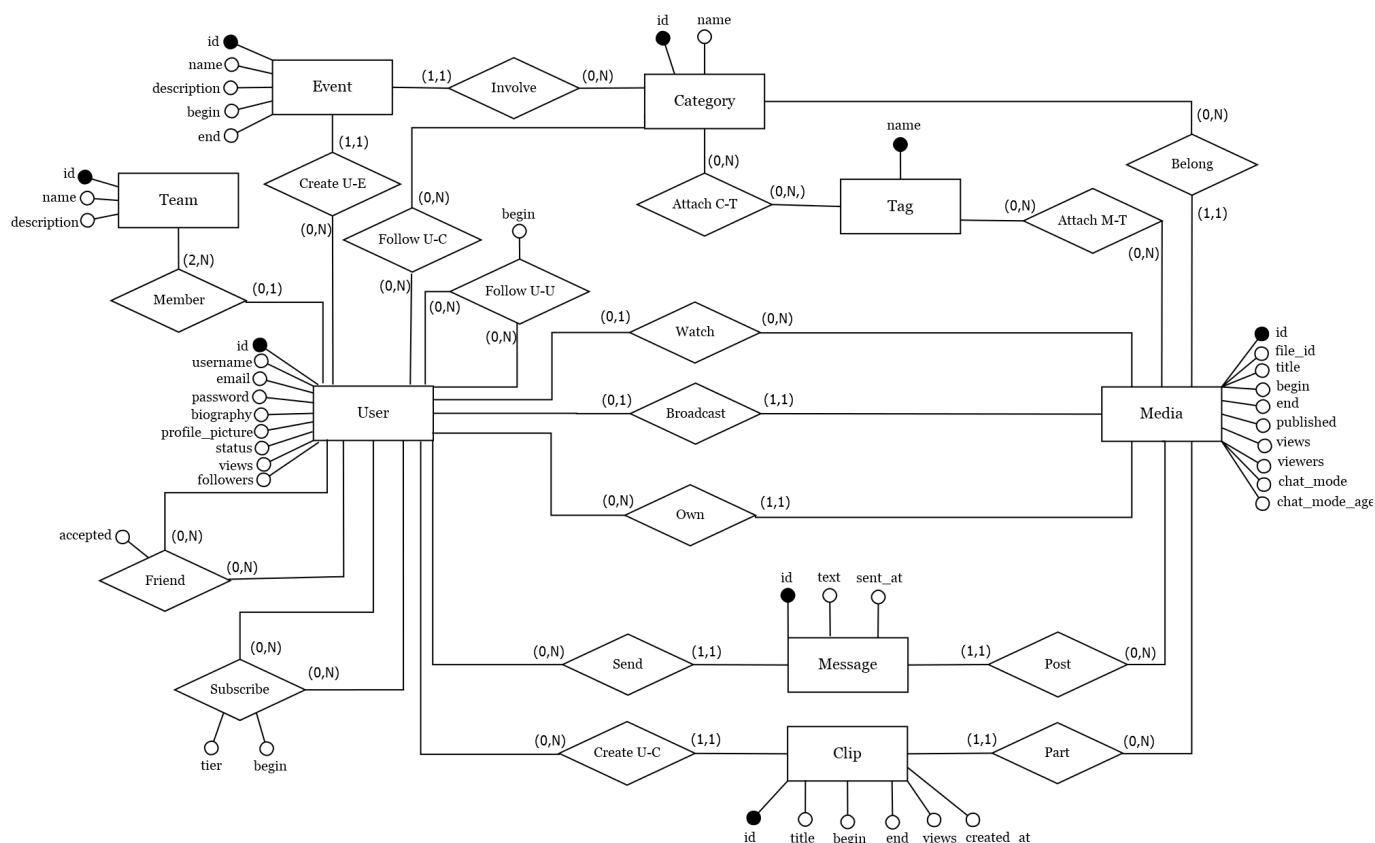
2.2.2 Lista entità

Entità	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
Livestream	Un video trasmesso in diretta live	viewers, chat_mode, chat_mode_age	id
Video	Una Livestream terminata	end, views, published	id
Message	Un messaggio testuale inviato nella chat di una Livestream	id, text, sent_at	id
Clip	Un frammento di una Livestream 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	Livestream programmata organizzata da un User	id, name, description, begin, end	id
Team	Gruppo uno o più utenti che gareggiano allo stesso gioco	id, name, description	id

2.2.3 Lista relazioni

Relazione	Descrizione	Cardinalità	Attributi
Watch	Associa un User alla Livestream che sta guardando	User(0, 1) Livestream(0, N)	
Broadcast	Associa un User alla Livestream 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	Livestream(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)	
Involve	Associa un Event alla Category di appartenenza	Event(1, 1) Category(0, N)	

2.3 Ristrutturazione



Nota Considerando che non è possibile rappresentare una generalizzazione in un normale DBMS, è stato necessario ristrutturare la gerarchia Media-Video-Livestream utilizzando il metodo dell'accorpamento. Tale metodo prevede che ogni attributo caratteristico delle entità figlie (Video, Livestream) venga trasferito all'entità genitore (Media). Per distinguere un'istanza di Video da una di Livestream è sufficiente analizzare il campo `end`: se `NULL` l'istanza è una Livestream, altrimenti è un Video.

2.3.1 Ridondanze

Sono state individuate diverse ridondanze su attributi derivabili da altre entità.

Concetto	Costrutto	Volume
Utente	E	1.000.000
Category	E	100
Livestream	E	5.000
Watch	R	400.000
Follow U-U	R	20.000.000

Operazione 14: visualizzare il numero di spettatori di una livestream (4.000.000 volte al giorno).

Presenza di ridondanza:

Concetto	Costrutto	Accesso	Tipo
Livestream	E	1	L

L'operazione viene eseguita 4M volte al giorno con un accesso in lettura ciascuna.

Costo: $4M \times 1 = 4M$ accessi giornalieri

Senza ridondanza:

Concetto	Costrutto	Accesso	Tipo
Utente	E	1	L
Livestream	E	1	L
Watch	R	100	L

Consideriamo che una livestream abbia una media di 100 spettatori.
 L'operazione viene eseguita 4M volte al giorno con 102 accessi in lettura ciascuna.
 Costo: $4M \times (100 + 1 + 1) = 408M$ accessi giornalieri

In conclusione, si preferisce tenere la ridondanza.

Operazione 15: visualizzare il numero totale di spettatori di tutte livestream per una data categoria (10.000.000 volte al giorno).

Presenza di ridondanza:

Concetto	Costrutto	Accesso	Tipo
Categoria	E	1	L

L'operazione viene eseguita 5M volte al giorno con un accesso in lettura ciascuna.
 Costo: $10M \times 1 = 10M$ accessi giornalieri

Senza ridondanza:

Concetto	Costrutto	Accesso	Tipo
Categoria	E	1	L
Livestream	E	5	L

Questa operazione può basarsi sull'operazione 14 in cui viene calcolato il numero di spettatori di una certa livestream.

Si suppone che in media ci siano 5 livestream per categoria.
 L'operazione viene eseguita 10M volte al giorno con 6 accessi ciascuna.
 Costo: $10M \times (5 + 1) = 60M$ accessi giornalieri

In conclusione, si preferisce eliminare la ridondanza.

Operazione 16: visualizzare il numero di followers di un utente (5.000.000 volte al giorno).

Presenza di ridondanza:

Concetto	Costrutto	Accesso	Tipo
Utente	E	1	L

L'operazione viene eseguita 5M volte al giorno con un accesso ciascuna.
 Costo: $5M \times 1 = 5M$ accessi giornalieri

Senza ridondanza:

Concetto	Costrutto	Accesso	Tipo
Utente	E	1	L
Follow U-U	R	10.0000	L

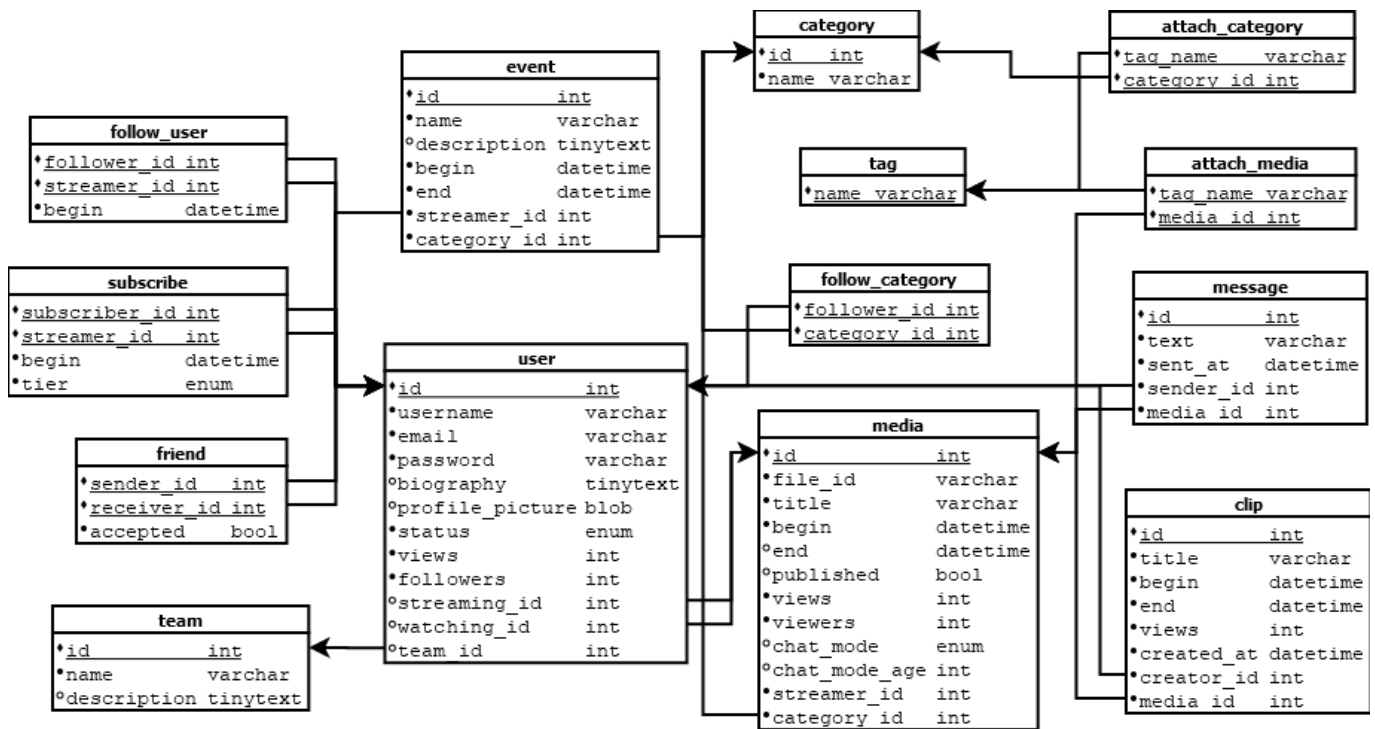
Si suppone che un utente sia più interessato a sapere il numero di follower degli utenti che trasmettono con più frequenza live e che quest'ultimi abbiano una media di 10K seguaci.

L'operazione viene eseguita 5M volte al giorno con 10.001 accessi ciascuna.
 Costo: $5M \times (10K + 1) = 50.005M$ accessi giornalieri

In conclusione, si preferisce tenere la ridondanza.

3 Progettazione logica

3.1 Implementazione dello schema logico



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)

3.2 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 Implementazioni

4.1 Query e procedure

Operazione 2

```
1  -- concludere una live ed eventualmente pubblicarla come video
2  DROP PROCEDURE IF EXISTS end_live;
3  DELIMITER |
4  CREATE PROCEDURE end_live(IN media_id INT, IN publish BOOL)
5  BEGIN
6      UPDATE media SET end = NOW(), published = publish
7      WHERE id = media_id;
8
9      UPDATE user SET watching_id = NULL
10     WHERE watching_id = media_id;
11  END |
12  DELIMITER ;
```

Operazione 15

```
1  -- visualizzare il numero totale di spettatori di tutte livestream per una data categoria
2  DELIMITER |
3  CREATE PROCEDURE category_viewers(IN category_id INT)
4  BEGIN
5      SELECT c.id, c.name, SUM(m.viewers) AS views
6      FROM category c INNER JOIN media m ON m.category_id = c.id WHERE c.id = category_id
7      GROUP BY c.id;
8  END |
9  DELIMITER ;
```

```
+---+-----+
| id | name       | views |
+---+-----+
|  6 | Super Mario |     4 |
+---+-----+
```

Operazione 18

```

1  -- visualizzare le live che stanno guardando gli amici di un utente
2  DELIMITER |
3  CREATE PROCEDURE search_friends_watching(IN user_id INT)
4  BEGIN
5      SELECT u.username AS friend_username, s.username AS watching_username, l.id AS live_id, c.id AS category_id, c.
        name AS category_name FROM user u
6      INNER JOIN
7          (SELECT f.receiver_id AS friend_id
8           FROM friend f
9           WHERE f.sender_id = user_id AND accepted = TRUE
10          UNION
11          SELECT f.sender_id AS friend_id
12           FROM friend f
13           WHERE f.receiver_id = user_id AND accepted = TRUE) f ON u.id = f.friend_id
14      INNER JOIN media l ON u.watching_id = l.id
15      INNER JOIN user s ON s.id = l.streamer_id
16      INNER JOIN category c ON l.category_id = c.id
17      WHERE u.status = 'online';
18  END |
19  DELIMITER ;

```

friend_username	watching_username	live_id	category_id	category_name
grimel	gamesoup	1	6	Super Mario
aleale	happymark	3	6	Super Mario

Operazione 19

```

1  -- ricercare livestream o video pubblicati che contengono un certo tag o una certa parola nel titolo
2  DELIMITER |
3  CREATE PROCEDURE search_media(IN word VARCHAR(128))
4  BEGIN
5      SELECT m.id, m.title
6      FROM media m
7      WHERE (m.end IS NULL OR m.published = TRUE) AND (m.title LIKE CONCAT('%', word, '%') OR word IN (SELECT a.
        tag_name FROM attach_media a WHERE a.media_id = m.id));
8  END |
9  DELIMITER ;

```

id	title
1	My new game
2	WORST GAME EVER
4	MY BORING PODCAST

Operazione 21

```

1  -- visualizzare le top 10 livestream con il maggior numero di spettatori, relative agli streamer seguiti dall'utente
2  DELIMITER |
3  CREATE PROCEDURE search_top_followed_live(IN user_id INT)
4  BEGIN
5      SELECT u.username AS streamer_username, l.id, l.title, l.viewers FROM media l
6      INNER JOIN follow_user f ON (f.follower_id = user_id AND f.streamer_id = l.streamer_id)
7      INNER JOIN user u ON u.id = l.streamer_id
8      WHERE l.end IS NULL
9      ORDER BY l.viewers LIMIT 10;
10  END |
11  DELIMITER ;

```

streamer_username	id	title	viewers
happymark	3	EPIC	2
mlime	4	MY BORING PODCAST	1

Operazione 22

```
1  -- visualizzare le live con il maggior numero di spettatori per ogni categoria
2  CREATE VIEW search_top_live_by_category AS
3      SELECT m.title, c.name AS category_name
4      FROM media m INNER JOIN category c ON c.id = m.category_id
5      WHERE m.end IS NULL AND m.id = (SELECT e.id FROM media e WHERE e.category_id = m.category_id ORDER by viewers
        DESC LIMIT 1);
```

```
+-----+-----+
| title           | category_name |
+-----+-----+
| My new game     | Super Mario   |
| MY BORING PODCAST | Podcast       |
+-----+-----+
```

4.2 Funzioni

Operazione 4

```
1  -- mandare, se permesso, un messaggio nella chat di una livestream
2  DELIMITER |
3  CREATE FUNCTION post_message(user_id INT, live_id INT, text VARCHAR(128)) RETURNS BOOL
4  BEGIN
5      DECLARE media_end DATETIME;
6      DECLARE active_chat_mode VARCHAR(20);
7      DECLARE active_chat_mode_age INT;
8      DECLARE allowed BOOL;
9      DECLARE since DATETIME;
10     DECLARE current_streamer_id INT;
11     SET since = NULL;
12     SET allowed = TRUE;
13     SELECT m.streamer_id INTO current_streamer_id FROM media m WHERE m.id = live_id;
14     SELECT m.end INTO media_end FROM media m WHERE m.id = live_id;
15     IF (media_end IS NULL AND current_streamer_id IS NOT NULL) THEN
16         SELECT m.chat_mode, m.chat_mode_age INTO active_chat_mode, active_chat_mode_age FROM media m WHERE m.id =
            live_id;
17         IF (active_chat_mode IS NOT NULL) THEN
18             IF (active_chat_mode = 'follow_only') THEN
19                 SELECT f.begin INTO since FROM follow_user f WHERE f.follower_id = user_id AND f.streamer_id =
                    current_streamer_id;
20             ELSE
21                 SELECT s.begin INTO since FROM subscribe s WHERE s.subscriber_id = user_id AND s.streamer_id =
                    current_streamer_id;
22             END IF;
23             IF (since IS NOT NULL) THEN
24                 IF (active_chat_mode_age IS NOT NULL AND UNIX_TIMESTAMP(NOW()) - UNIX_TIMESTAMP(since) <
                    active_chat_mode_age) THEN
25                     SET allowed = FALSE;
26                 END IF;
27             ELSE
28                 SET allowed = FALSE;
29             END IF;
30         END IF;
31     ELSE
32         SET allowed = FALSE;
33     END IF;
34     IF (allowed) THEN
35         INSERT INTO message(text, sender_id, media_id) VALUES(text, user_id, live_id);
36     END IF;
37     RETURN allowed;
38 END |
39 DELIMITER ;
```

Operazione 11

```
1  -- mandare richieste di amicizia ad altri utenti
2  DELIMITER |
3  CREATE FUNCTION send_friend_request(s_id INT, r_id INT) RETURNS BOOL
4  BEGIN
5      DECLARE already_exists INT;
6      SELECT sender_id INTO already_exists FROM friend f WHERE (f.sender_id = s_id AND f.receiver_id = r_id)
7      OR (f.sender_id = r_id AND f.receiver_id = s_id);
8
9      IF(already_exists IS NOT NULL) THEN
10         return FALSE;
11     ELSE
12         INSERT INTO friend(sender_id, receiver_id) VALUES(s_id, r_id);
13         return TRUE;
14     END IF;
15 END |
16 DELIMITER ;
```

4.3 Trigger

Regola di vincolo

```
1  -- il contatore viewers di una livestream deve essere aggiornato quando un utente inizia o finisce di guardarla
2  DELIMITER |
3  CREATE TRIGGER update_watching_counter
4  AFTER UPDATE ON user
5  FOR EACH ROW
6  BEGIN
7      DECLARE old_counter INT;
8      IF (NEW.watching_id <> OLD.watching_id OR NEW.watching_id IS NULL OR OLD.watching_id IS NULL) THEN
9          IF(OLD.watching_id IS NOT NULL) THEN
10             SELECT m.viewers INTO old_counter
11             FROM media m
12             WHERE m.id = OLD.watching_id;
13             UPDATE media SET viewers = old_counter - 1
14             WHERE id = old.watching_id;
15         END IF;
16         IF (NEW.watching_id IS NOT NULL) THEN
17             SELECT m.viewers INTO old_counter
18             FROM media m
19             WHERE m.id = new.watching_id;
20             UPDATE media SET viewers = old_counter + 1
21             WHERE id = new.watching_id;
22         END IF;
23     END IF;
24 END |
25 DELIMITER ;
```

Regola di vincolo

```
1  -- il contatore followers di un utente deve essere aggiornato quando quest'ultimo acquisisce o perde un seguace
2  DELIMITER |
3  CREATE TRIGGER increase_followers_counter
4  AFTER INSERT ON follow_user
5  FOR EACH ROW
6  BEGIN
7      DECLARE old_counter INT;
8      SELECT followers INTO old_counter
9      FROM user u WHERE u.id = NEW.streamer_id;
10     UPDATE user SET followers = old_counter + 1
11     WHERE id = NEW.streamer_id;
12 END |
13 DELIMITER ;
```

Regola di vincolo

```
1  -- il contatore followers di un utente deve essere aggiornato quando quest'ultimo acquisisce o perde un seguace
2  DELIMITER |
3  CREATE TRIGGER decrease_followers_counter
4  AFTER DELETE ON follow_user
5  FOR EACH ROW
6  BEGIN
7      DECLARE old_counter INT;
8      SELECT followers INTO old_counter
9      FROM user u WHERE u.id = OLD.streamer_id;
10     UPDATE user SET followers = old_counter - 1
11     WHERE id = OLD.streamer_id;
12 END |
13 DELIMITER ;
```

Nota Sono state implementate come query, procedures, o funzioni, solo le operazioni più significative.