

Boscaglia Pierluigi, Cascione Michele
PROGETTO DATABASE
ANNO 2023

1.1 Requisiti Iniziali

Si vuole realizzare una base di dati per un servizio che permette di fare live streaming su vari argomenti. Il live streaming (o, più sinteticamente, la live) permette di interagire con il pubblico in tempo reale grazie a feed video, chat e altro. Ogni utente può essere spettatore o streamer, o entrambi. Gli spettatori possono essere registrati al servizio oppure possono guardare le live in modo anonimo. Per registrarsi, gli utenti devono indicare nome utente, password, data di nascita, numero di telefono o indirizzo mail. Gli utenti iscritti possono chattare, seguire lo streamer, creare dirette. Gli streamer hanno ciascuno un canale, che può essere caratterizzato tramite una descrizione. Per ogni canale, è possibile specificare una lista di social associati (ad esempio Instagram, YouTube, ecc.), un'immagine profilo e anche un trailer (Figura 1(a)). In ogni canale possono esserci live, video (live passate) e clip (video di durata breve). Le live possono anche non diventare video del canale. Ognuno ha un titolo, una durata, appartiene a una categoria (Figura 1(b)) e può essere associato a diversi tag. Per ogni live viene memorizzato il numero medio di spettatori mentre per i video e le clip il numero di visualizzazioni.

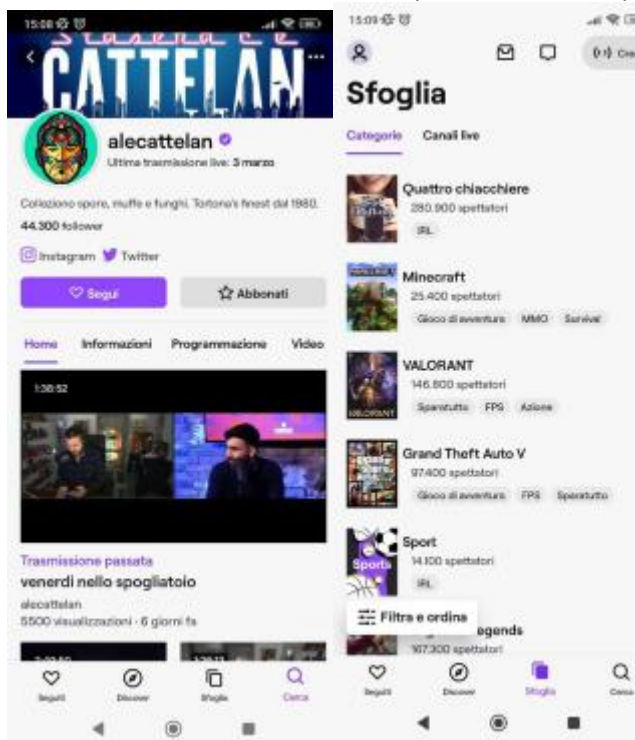


Figura 1:

- (a) La home del canale di alecattelan
- (b) La lista delle categorie

Per ogni creatore di contenuti, si memorizzano il numero di live effettuate, il numero di minuti trasmessi, il numero medio di spettatori simultanei. Inoltre, sulla pagina del canale viene visualizzato il numero di follower. Quando uno streamer rispetta determinati parametri di performance (un minimo di 500 minuti trasmessi, una media di tre o più spettatori simultanei, almeno 50 follower), può diventare affiliate. Le stream hanno degli orari. Ogni streamer ha un calendario in cui può dire quando farà stream e indicare il titolo delle prossime live. I viewer possono diventare follower del canale degli streamer che preferiscono, e le loro preferenze sono raccolte in un elenco di followee a cui possono accedere dal loro profilo. I viewer possono inoltre supportare gli streamer tramite la subscription (a pagamento) al loro canale, ottenendo dei privilegi (emoticon personalizzate, ecc.). Inoltre, gli utenti hanno un portafoglio di bit (moneta virtuale che possono acquistare tramite la piattaforma), che possono usare per fare donazioni agli streamer.

Oltre a chattare pubblicamente, gli utenti possono scambiarsi messaggi privati. La base di dati deve supportare le seguenti operazioni:

- Una volta al giorno si controllano le condizioni per la qualifica di affiliate.
- Una volta a settimana viene calcolata la classifica degli streamer più seguiti. Si può assumere che i contenuti multimediali vengano gestiti da una piattaforma di video hosting e che quindi sia sufficiente memorizzare un URL.

1.2 Glossario:

TERMINE	DESCRIZIONE	SINONIMI	COLLEGAMENTI
Live	Trasmissione streaming iniziata dall'utente	Live Streaming, Stream, Dirette	Utente, Canale, Video
Utente	Persona che accede alla piattaforma streaming	Pubblico, Spettatore, Streamer, Creatore di contenuti, Viewer	Canale, Video, Subscription, Donazione, Calendario
Canale	Pagina dedicata agli utenti streamer	Pagina del canale	Live, Video, Utente, Subscription
Video	Contenuto multimediale presente sul canale dello streamer	Live passate, Clip	Live, Utente
Donazione	Pagamento in bit diretto dall'utente al canale	//	Utente, canale
Subscription	Sottoscrizione a pagamento diretta al canale da un utente	//	Utente, Canale
Calendario	Contiene data e titolo delle prossime live che farà l'utente streamer	//	Utente, Calendario
Messaggio (messaggiare)	Testo inviato in chat live o tra due utenti in chat privata	Chattare, Scambiarsi messaggi privati	Utente, Live

1.3 Requisiti Rivisti:

Si vuole realizzare una base di dati per un servizio che permette di fare live streaming su vari argomenti. La live permette di interagire con gli utenti in tempo reale grazie a feed video, chat e altro. Gli utenti possono essere registrati al servizio oppure possono guardare le live in modo anonimo. Per registrarsi, gli utenti devono indicare nome utente, password, data di nascita, numero di telefono o indirizzo mail. Gli utenti iscritti possono chattare, seguire altri utenti e creare live. Gli utenti hanno ciascuno un canale, che può essere caratterizzato tramite una descrizione. Per ogni canale, è possibile specificare una lista di social associati (ad esempio Instagram, YouTube, ecc.), un'immagine profilo e anche un trailer (Figura 1(a)). In ogni canale possono esserci live e contenuti multimediali. Ogni contenuto multimediale ha un titolo, una durata, appartiene a una categoria (Figura 1(b)) e può essere associato a diversi tag. Le live possono anche non diventare contenuti multimediali del canale. Per ogni live viene memorizzato il numero medio di utenti, per i contenuti multimediali viene memorizzato il numero di visualizzazioni. Per ogni utente, si memorizzano il numero di live effettuate, il numero di minuti trasmessi, il numero medio di utenti simultanei. Inoltre, sul canale viene visualizzato il numero di follower. Quando un utente rispetta determinati parametri di performance (un minimo di 500 minuti trasmessi, una media di tre o più utenti simultanei, almeno 50 follower), può diventare affiliate. Le live hanno degli orari. Ogni utente ha un calendario in cui può dire quando farà live e indicare il titolo delle prossime live. Gli utenti possono diventare follower del canale che preferiscono, e le loro preferenze sono raccolte in un elenco di followee a cui possono accedere dal loro profilo. Gli utenti possono inoltre supportarne altri tramite la subscription (a pagamento) al loro canale, ottenendo dei privilegi (emoticon personalizzate, ecc.). Inoltre, gli utenti hanno un portafoglio di bit (moneta virtuale che possono acquistare tramite la piattaforma), che possono usare per fare donazioni agli utenti. Oltre a messaggiare pubblicamente, gli utenti possono messaggiare

privatamente. La base di dati deve supportare le seguenti operazioni:

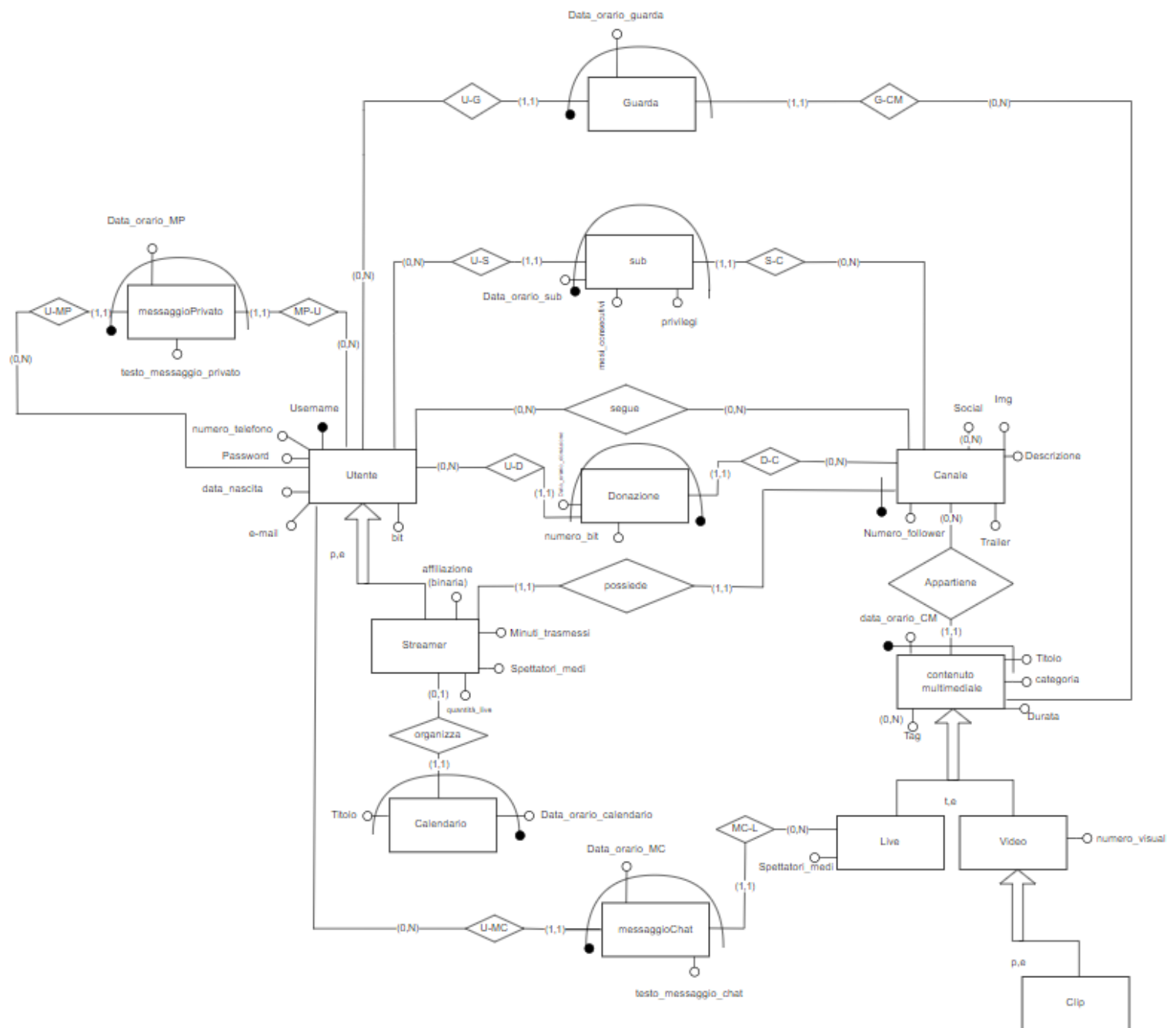
- Una volta al giorno si controllano le condizioni per la qualifica di affiliate.
- Una volta a settimana viene calcolata la classifica degli utenti più seguiti. Si può assumere che i video vengano gestiti da una piattaforma di video hosting e che quindi sia sufficiente memorizzare un URL.

Frasi ristrutturate:

- **Frasi di carattere generale:** Si vuole realizzare una base di dati per un servizio che permette di fare live streaming su vari argomenti;
- **LIVE:** Per ogni live viene memorizzato il numero medio di utenti e permettono di interagire in tempo reale grazie a feed video, chat e altro;
- **UTENTE:** Gli utenti hanno un nome utente, password, data di nascita, numero di telefono o indirizzo mail. Se iscritti possono chattare, seguire e creare live. Possono essere registrati al servizio o guardare le live in modo anonimo. Per ognuno si memorizzano il numero di live effettuate, il numero di minuti trasmessi, il numero medio di utenti simultanei. Quando rispetta determinati parametri di performance (un minimo di 500 minuti trasmessi, una media di tre o più utenti simultanei, almeno 50 follower), l'utente diventa affiliate. Gli utenti possono diventare follower del canale che preferiscono, e le loro preferenze sono raccolte in un elenco di followee a cui possono accedere dal loro profilo. Una volta a settimana viene calcolata la classifica degli utenti più seguiti;
- **CANALE:** Gli utenti hanno ciascuno un canale caratterizzato da una descrizione, una lista di social associati (ad esempio Instagram, YouTube, ecc.), un'immagine profilo, il numero di follower, live, contenuti multimediali e un trailer (Figura 1(a));
- **VIDEO:** Ogni contenuto multimediale ha un titolo, una durata, il numero medio di visualizzazioni, appartiene ad una categoria (Figura 1(b)) e può essere associato a diversi tag. Le live possono anche non necessariamente farne parte. Si può assumere che i video vengano gestiti da una piattaforma di video hosting;
- **DONAZIONE:** Gli utenti hanno un portafoglio di bit (moneta virtuale che possono acquistare tramite la piattaforma), che possono usare per fare donazioni;
- **SUBSCRIPTION:** Gli utenti possono inoltre supportarne altri tramite la subscription (a pagamento) al loro canale, ottenendo dei privilegi (emoticon personalizzate, ecc.);
- **CALENDARIO:** Il calendario permette di rappresentare data e titolo delle prossime live.
- **MESSAGGIO (MESSAGGIARE):** Oltre a messaggiare pubblicamente, gli utenti possono messaggiare privatamente.

1.4 Schema E-R principale + business rules

1.4.1 Modello relazionale



Modello sviluppato con l'utilizzo di Draw.io

https://drive.google.com/file/d/1Q6U5crG9ImZ2-csxUPNoO9nBtRaPJRCy/view?usp=drive_link

1.4.2 Business rule descrittiva

Tabella delle entità

<i>Entità</i>	<i>Descrizione</i>	<i>Attributi</i>	<i>Identificatore</i>
Utente	utenti che usufruiscono dei servizi del sito	Username, numero_telefono, password, data_nascita, e-mail, bit	Username
Streamer	categoria di utenti che effettua live sulla piattaforma	Username, numero_telefono, password, data_nascita, e-mail, bit, affiliazione, minuti_trasmessi, spettatori_medi	Username
Calendario	calendario utilizzato dallo streamer per segnare eventi	Data_orario_calendario, titolo	Username, Data_orario_calendario, titolo
Donazione	Donazione di bit che viene fatta al canale	Numero_bit, Data_orario_donazione	Username, Data_orario_donazione
Canale	Il canale posseduto da ogni streamer	numero_follower, social, img, descrizione, trailer	Username
Contenuto multimediale	è ciò che un utente può generare attraverso il canale	Data_orario_CM, titolo, categoria, durata, numero_visualizzazioni, Tag	Data_orario_CM, titolo, Username
Live	è un contenuto multimediale trasmesso in diretta	Data_orario_CM, titolo, categoria, durata, numero_visualizzazioni, Tag, spettatori_medi	Data_orario_CM, titolo, Username
Video	è un contenuto multimediale che rappresenta una diretta passata	Data_orario_CM, titolo, categoria, durata, numero_visualizzazioni, Tag, numero_visual	Data_orario_CM, titolo, Username
Clip	sono spezzoni corti di video generati dagli utenti	Data_orario_CM, titolo, categoria, durata, numero_visualizzazioni, Tag	Data_orario_CM, titolo, Username
Sub	abbonamento di un utente al canale di uno streamer	mesi_consecutivi, data_orario_sub, privilegi	Data_orario_sub, Username, Username

Guarda	rappresentazione di un utente che guarda un contenuto multimediale	Data_orario_guarda	Data_orario_guarda, Username, titolo, Data_orario_CM, <u>Username</u>
MessaggioPrivato	Messaggio che un utente invia privatamente ad un altro utente	Data_orario_MP	Data_orario_MP, Username, Username
MessaggioChat	Messaggio che un utente invia nella chat di una live	Data_orario_MC	Data_orario_MP, Username, Data_orario_CM, Titolo,

Tabella delle relazioni

Relazione	Descrizione	Entità	Attributi
segue	Associa un utente ad un canale	Utente (0,N) Canal (0,N)	
possiede	Associa uno streamer ad un canale	Streamer (1,1) Canale(1,1)	
U-S	Associa un utente ad una sub effettuata	Utente (0,N) Sub (1,1)	
S-C	Associa una sub ad un canale	Sub (1,1) Canale (0,N)	
U-G	Associa utente a guarda	Utente (0,N) Guarda (1,1)	
G-CM	Associa il guarda a contenuto multimediale	Guarda (1,1) Contenuto multimediale (0,N)	
appartiene	Associa un canale a dei contenuti multimediali	Canale (0,N) Contenuto multimediale (1,1)	
U-MC	Associa utente a messaggio in chat	Utente (0,N) messaggioChat (1,1)	
MC-L	Associa messaggio in chat ad una live	messaggioChat (0,N) Live (1,1)	
organizza	Associa uno streamer ad un calendario	streamer (0,1) Calendario (1,1)	

D-C	Associa una donazione ad un canale	Donazione (1,1) Canale (0,N)	
U-D	Associa un utente ad una donazione	Utente (0,N) Donazione (1,1)	
U-MP	Associa utente a messaggio privato	Utente (0,N) messaggioPrivato (1,1)	
MP-U	Associa messaggio privato ad un utente	messaggioPrivato (1,1) Utente (0,N)	

1.4.3 Business rule non descrittiva

Regole di vincolo

RV1: Il numero di bit donati da un utente **non deve** superare il numero di bit che possiede

RV2: gli spettatori medi di una live **non devono** essere maggiori del numero di visualizzazioni del video

RV3: un utente può avere o una mail o un numero di telefono, non può averli entrambi

RV4: La sub scade dopo un mese dalla sua avvenuta, solo in quel momento sarà possibile per quel Utente effettuare l'abbonamento al canale

RV5: in **messaggioPrivato** bisogna aggiungere un vincolo in modo da non poter **permettere** che un **Utente scriva a se stesso**, ma comunichi solo con altri utenti, i due attributi **Username** di cui è composta la chiave **non possono quindi essere gli stessi**

RV7: gli attributi **TIMESTAMP** non possono essere modificati una volta che vengono settati

RV8: uno streamer **deve avere numero medio di spettatori, numero di minuti trasmessi e stream effettuare tutti maggiori di 0**

RV9: può essere **avviata una live per volta**, quindi non si può avviare un'altra live se prima non è **terminata**

RV10: I **privilegi non saranno più disponibili** dopo la **scadenza della sub** (ha durata **30 giorni**)

Regole di derivazione

RD1: L'affiliazione (true) si ottiene guardando: minuti trasmessi (superiori a 500), media di spettatori (superiore a 2), numero di follower (almeno 50)

RD2: gli spettatori medi si ottengono sommando ogni ora per n volte il numero di spettatori e dividendoli per n

RD3: Ogni occorrenza di Utente può partecipare a Streamer e ogni occorrenza di Streamer partecipa a Utente

RD4: Ogni occorrenza di Contenuto multimediale può partecipare o a Live o Video

RD5: Ogni occorrenza di Video può partecipare a Clip e ogni occorrenza di Clip partecipa a Video

2. Progettazione logica

2.1 Tavola dei volumi (con motivazione delle scelte effettuate)

CONCETTO	TIPO	VOLUME
Donazione	E	1MLD
messaggioPrivato	E	1MLD
Utente	E	1MLD
Streamer	E	200k
Calendario	E	7M
Guarda	E	50MLD
Sub	E	30M
messaggioChat	E	10MLD
Canale	E	200k
Contenuto Multimediale	E	10MLD
Live	E	1M
Video	E	4MLD
Clip	E	6MLD
U-MP	A	1MLD
MP-U	A	1MLD
U-D	A	1MLD
D-S	A	1MLD
Organizza	A	7M
U-G	A	50MLD
U-S	A	30M
Segue	A	5MLD
Possiede	A	10M
U-MC	A	10MLD
S-C	A	30M
G-CM	A	50MLD
MC-L	A	10 MLD
Appartengono	A	10MLD

Gli **streamer** in una piattaforma di tale importanza sono un numero molto elevato (200k); contando il fatto che chiunque abbia un canale è categorizzabile come “streamer”, allora gli **utenti** si aggirano su 1MLD. Il numero di **sub** è indicativamente 30M, questo perché un utente può abbonarsi a diversi streamer contemporaneamente ma non tutti gli utenti sono disposti a versare somme di denaro per utilizzare la piattaforma. Stesso discorso vale per le **donazioni** (1MLD), ma queste sono molto più frequenti delle sub e richiedono anche piccole quantità di denaro. Il numero di **canali** corrisponde al numero degli streamer (200k). Sono presenti circa 7M di **calendari**, questo perché non tutti gli streamer ne ritengono necessario l’uso (è possibile streammare anche senza programmare la live).

Date le diverse live trasmesse dagli streamer ogni giorno, i **contenuti multimediali** generati da queste giornalmente sono un numero elevato (10MLD). Le **live** vengono terminate nel momento in cui lo streamer decide di interrompere la stream. Mediamente le live che esistono nello stesso momento si aggirano sul 1M. I **video** rappresentano le live passate che sono state salvate; si può pensare che su 10MLD di contenuti multimediali, meno della metà siano live passate e il resto siano clip. Quindi i video si aggirano sui 4MLD, invece le **clip** sui 6MLD. In una live mediamente viene inviato un numero di messaggi (**messaggioChat**) molto elevato; considerando che questi vengono memorizzati per essere utilizzati nelle live salvate, si aggirano sui 10MLD. Dato l’uso ridotto delle chat private tra due utenti, i **messaggi privati** si aggirano su 1MLD. Quando un utente **guarda** un contenuto multimediale (quindi una live e/o una clip), il dato viene salvato ed è considerevolmente alto (50MLD).

Per quanto riguarda le associazioni, data la cardinalità specificata su ogni collegamento associazione-entità, la cardinalità coincide tra quest’ultime. Questo non si verifica nel caso dell’associazione segue: mediamente

ogni utente segue molti streamer e ogni streamer è seguito da molti utenti, quindi il volume di segue risulta essere elevato (5MLD).

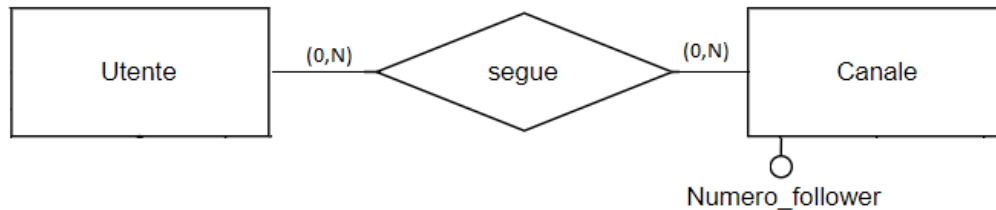
2.2 Tavola delle operazioni

DESCRIZIONE	TIPO	FREQUENZA
Un nuovo utente si registra alla piattaforma	I	100K al giorno
Un utente entra in live o guarda un contenuto multimediale	I	30M al giorno
Uno streamer inizia una live	I	100K a settimana
Viene salvata una live o una clip	I	270K a settimana
Viene memorizzato il numero medio di spettatori e di visualizzazioni per il contenuto multimediale	B	17M a settimana
Un utente segue uno streamer	I	5M al giorno
Viene aggiornato l'elenco dei followee	B	1M al giorno
Un utente si abbona ad un canale	I	
Uno streamer aggiorna il suo calendario	I	500K a settimana
Un utente scrive nella chat live	I	1MLD al giorno
Un utente messaggia un altro utente	I	30M al giorno
Si controlla se uno streamer è qualificato per essere affiliate	B	1 volta al giorno
Si calcola la classifica degli streamer più seguiti	B	1 volta a settimana

2.3 Ristrutturazione dello schema ER

2.3.1 Analisi delle ridondanze

Ridondanze rilevate: l'attributo "live effettuate" presenti sull'entità Streamer è derivabile dalla composizione delle associazioni Streamer-possiede-canale-appartiene-contenuto multimediale e l'attributo "Numero_follower" presente sull'entità "Canale" è derivabile dalla composizione dell'associazione Utente-



Segue-Canale.

Operazioni e volumi su cui la presenza/assenza di ridondanza ha effetto:

DESCRIZIONE	TIPO	FREQUENZA
Op1) Un utente segue uno streamer	I	5M al giorno
Op2) Si controlla se uno streamer è qualificato per essere affiliate	B	1 volta al giorno

CONCETTO	TIPO	VOLUME
Utente	E	1MLD
Canale	E	200k
Segue	A	5MLD

- Schema di navigazione e tavola degli accessi in assenza della ridondanza:

Operazione 1: Un utente segue uno streamer

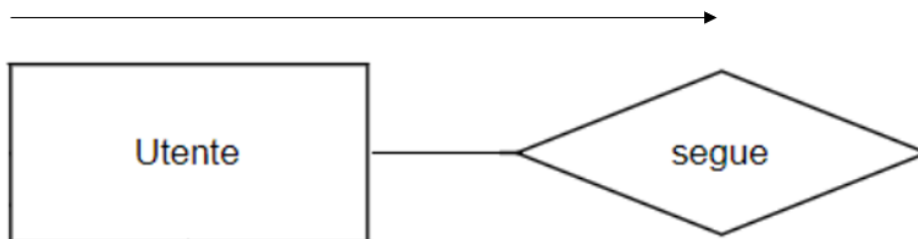


Tavola degli accessi:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Utente	Entità	1	S
Segue	Associazione	1	S

Operazione 2: Si controlla se uno streamer è qualificato per essere affiliate

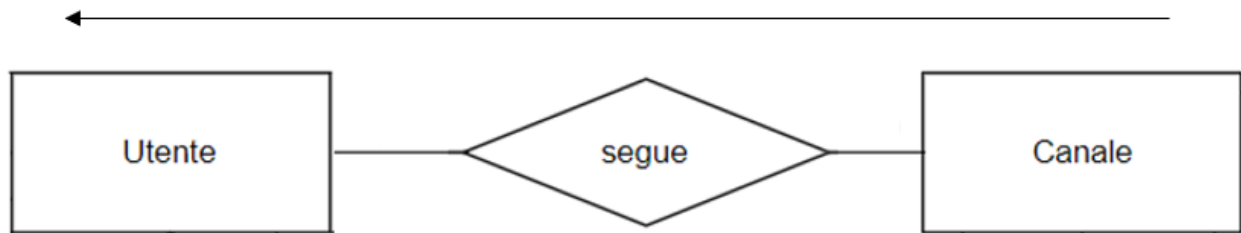


Tavola degli accessi:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Canale	Entità	1	L
Segue	Associazione	25000	L
Utente	Entità	1	L
Utente	Entità	1	L

- Schema di navigazione e tavola degli accessi in presenza della ridondanza:

Operazione 1: Un utente segue uno streamer

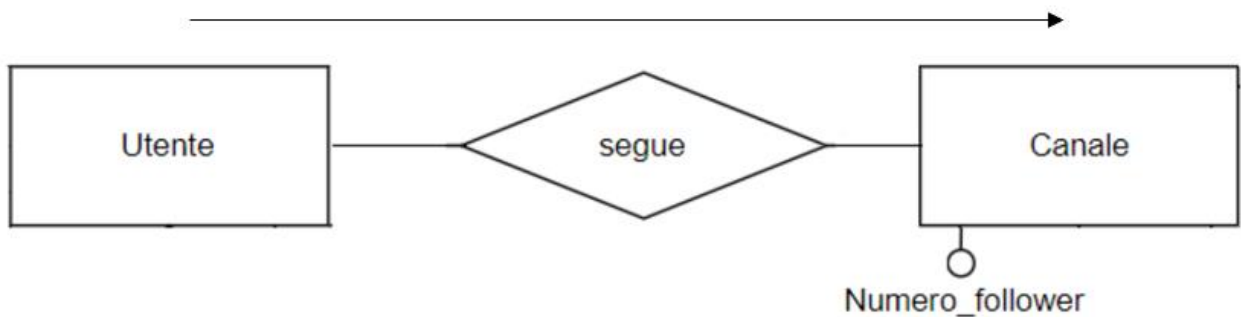


Tavola degli accessi:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Utente	Entità	1	S
Segue	Associazione	1	S
Canale	Entità	1	L
Canale	Entità	1	S

Con la ridondanza presente, dopo aver seguito un canale viene aggiornato il numero di follower di quest'ultimo.

Operazione 2: Si controlla se uno streamer è qualificato per essere affiliate

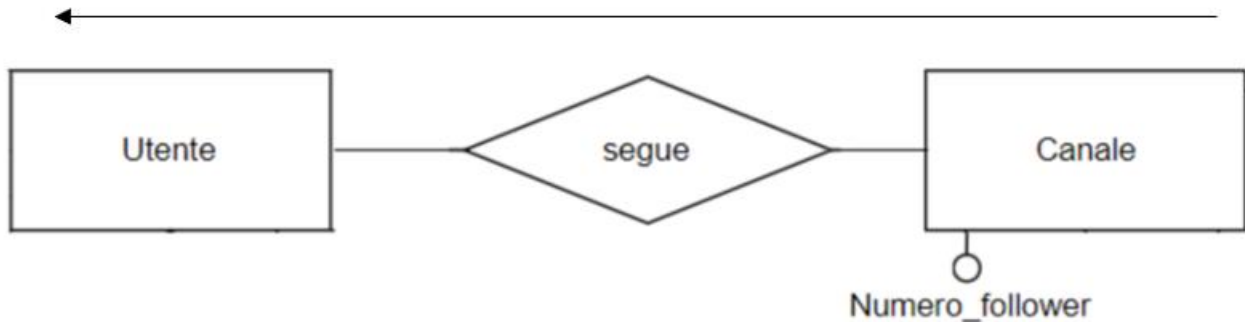


Tavola degli accessi:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Canale	Entità	1	L
Segue	Associazione	1	L
Utente	Entità	1	L
Utente	Entità	1	L

- Confronto in spazio e tempo tra presenza e assenza di ridondanza

Assenza della ridondanza:

Spazio: 0 byte.

Tempo:

- **Op1:** $2 * (2)5M$ di accessi in scrittura al giorno = **20M**;

- **Op2:** $25000 * 1$ accesso in lettura al giorno = **25000**.

Totale 20.025.000 accessi.

Presenza della ridondanza:

Spazio: 4 Byte per il numero di follower di un canale ($\sim 2^{32}$ follower): $4 \text{ Byte} * 10M \text{ Canali} = \mathbf{40 \text{ Mega Byte}}$.

Tempo:

- **Op1:** $3 * (2)5M$ di accessi in scrittura al giorno e $5M$ di accessi in lettura al giorno = **35M**;

- **Op2:** $4 * 1$ di accessi in lettura al giorno. = **4**.

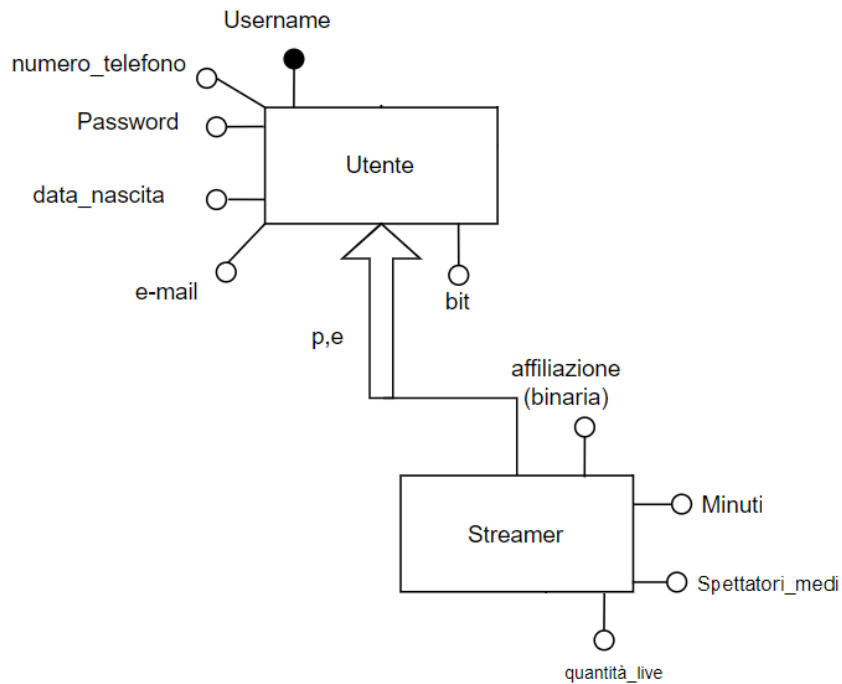
Totale 35.000.004 accessi.

- Scelta se introdurre o non introdurre la ridondanza:

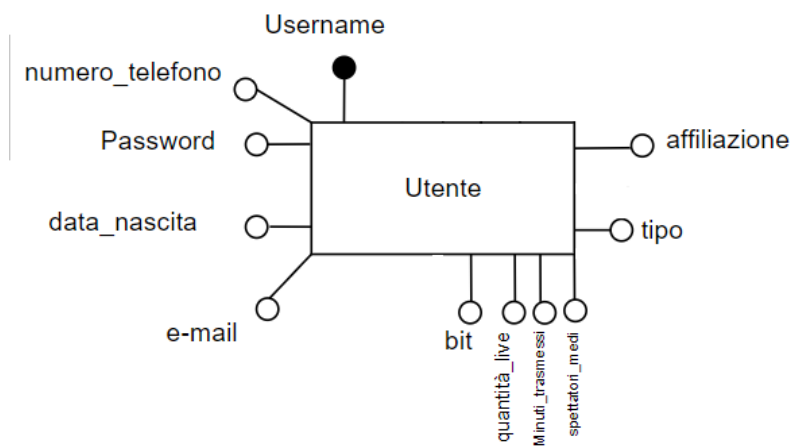
Data la notevole differenza negli accessi in assenza della ridondanza, è opportuno **non introdurla**. Il dato risulta essere di comodo utilizzo nell'operazione 2, perchè ci permette di eliminare i 25000 accessi. Il problema è presente, però, sull'operazione 1: aggiornare il dato ogni volta che viene seguito un canale richiede un'operazione in lettura ed una in scrittura. L'operazione viene eseguita milioni di volte al giorno, quindi tutti gli accessi in scrittura per aggiornare il dato vanno ad aumentare di molto gli accessi totali.

2.3.2 Eliminazioni delle generalizzazioni (con motivazioni)

Caso 1

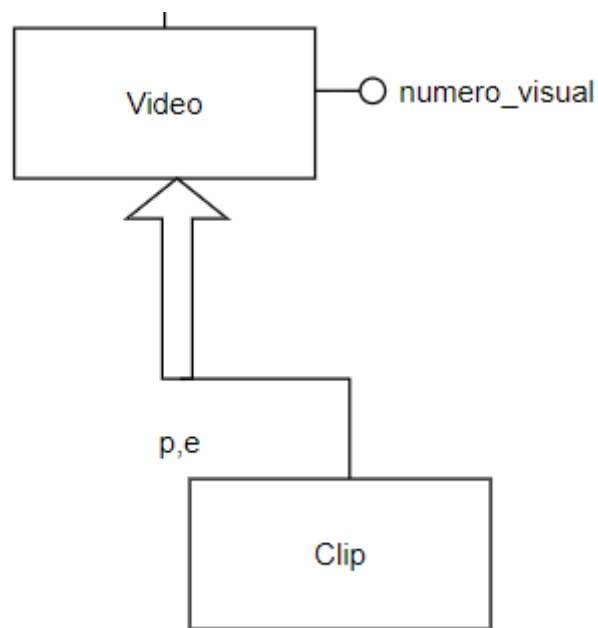


la prima generalizzazione da eliminare è quella parziale esclusiva di Utente che può essere anche uno Streamer, quest'ultimo eredita tutti gli attributi di Utente e in più ne possiede 4 che appartengono solamente a lui, abbiamo scelto di eliminare la generalizzazione effettuando un **accorpamento dell'entità figlia a quella genitore**, verrà gestita con l'aggiunta dell'attributo **tipo** in Utente e gli attributi che con la generalizzazione appartenevano a Streamer avranno una **cardinalità(0,1)** a seconda del tipo di **Utente**.

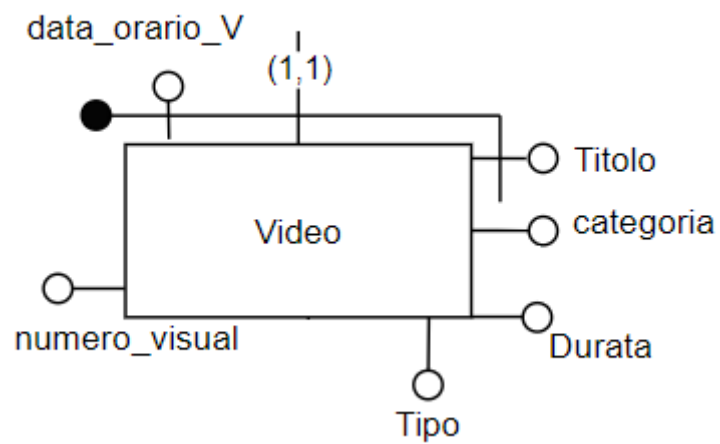


La relazione da Streamer a canale che prima vedeva Streamer come **cardinalità (1,1)** dopo l'accorpamento avrà **cardinalità (0,1)**.

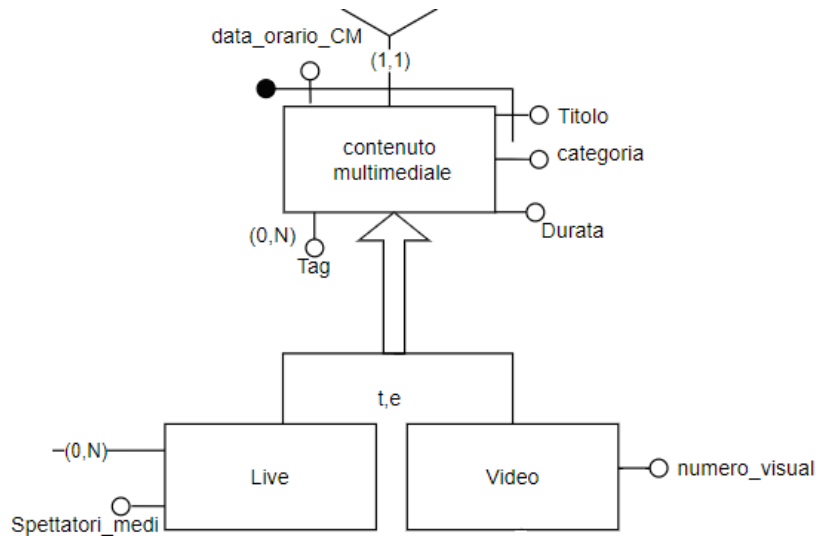
Caso 2



La seconda generalizzazione è più complicata, in primo luogo vi è una generalizzazione tra **Video-Clip** che è come quella vista in precedenza nel caso **Utente-Streamer**, quindi inseriamo un attributo **tipo** a **Video** in modo da distinguere le clip dai video, visto che **Clip** non ha attributi ovviamente non verrà aggiunto altro.

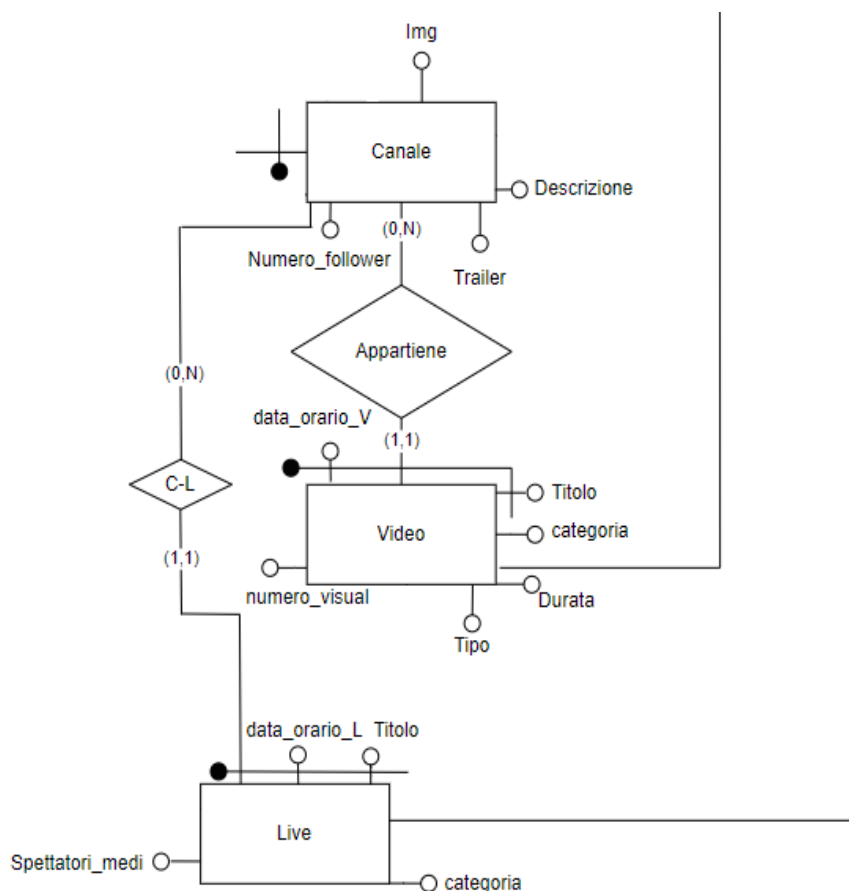


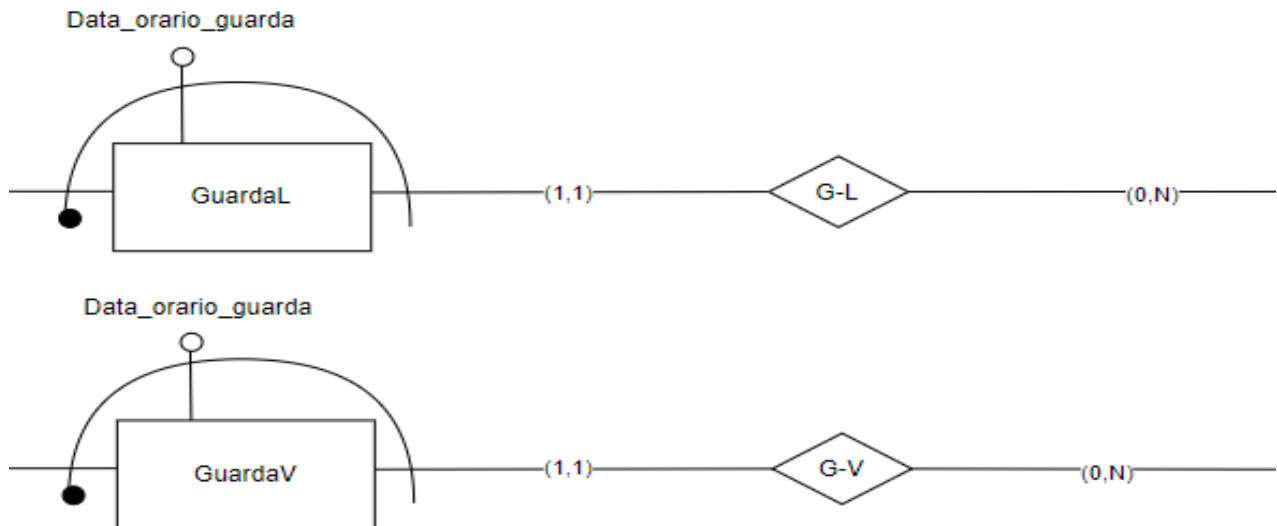
Caso 3



Per quanto riguarda la generalizzazione superiore quindi quella che riguarda i **contenuto multimediale**, visto che si tratta di una generalizzazione **totale/esclusiva** elimineremo la generalizzazione aggiungendo **due entità**, una per **Video** e una per **Live**, in entrambi i casi, ovviamente bisogna aggiungere sia alla nuova entità **Video** che a **Live** le relazioni che prima riguardavano **contenuto multimediale**, nel caso una relazione precedente riguardasse un'entità figlie verrà semplicemente mantenuta.

Gli attributi che erano inseriti prima in **contenuto multimediale** andranno aggiunte ad entrambe le entità.



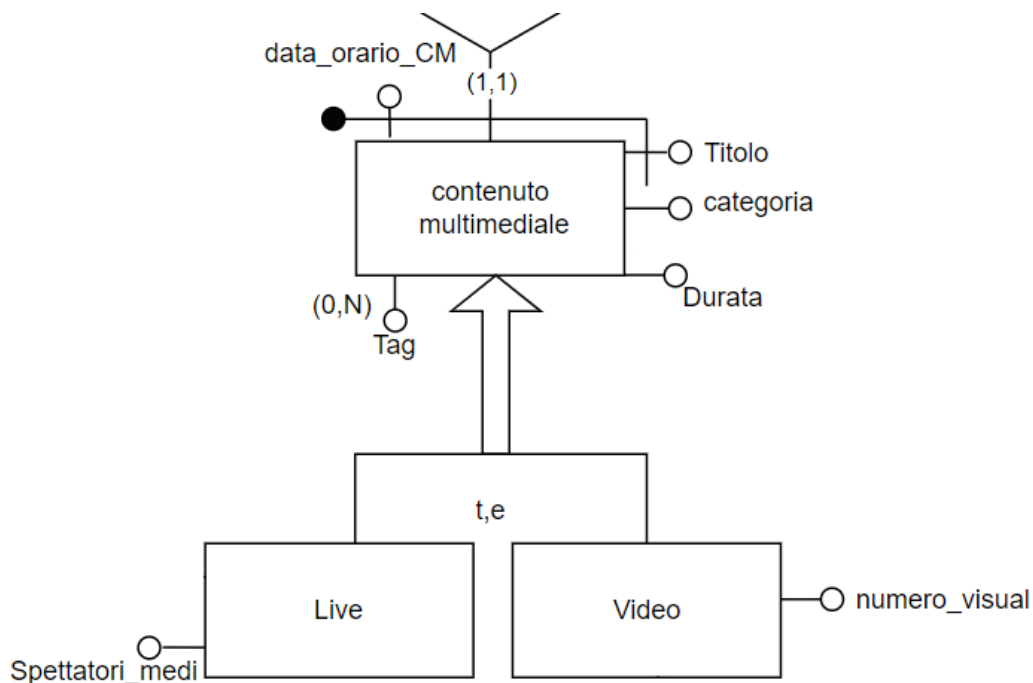


Avendo una relazione (0,N) (0,N) tra **contenuto multimediale** e **Utente** abbiamo dovuto creare una distinzione per il **Guarda** che fa riferimento all'entità figlio **Video** (**GuardaV**) e quella riferita a **Live** (**GuardaL**)

2.3.4 Eliminazione degli attributi composti e degli attributi multivalore.

Per l'eliminazione degli attributi multivalore semplicemente trasformiamo l'attributo in questione in una nuova entità

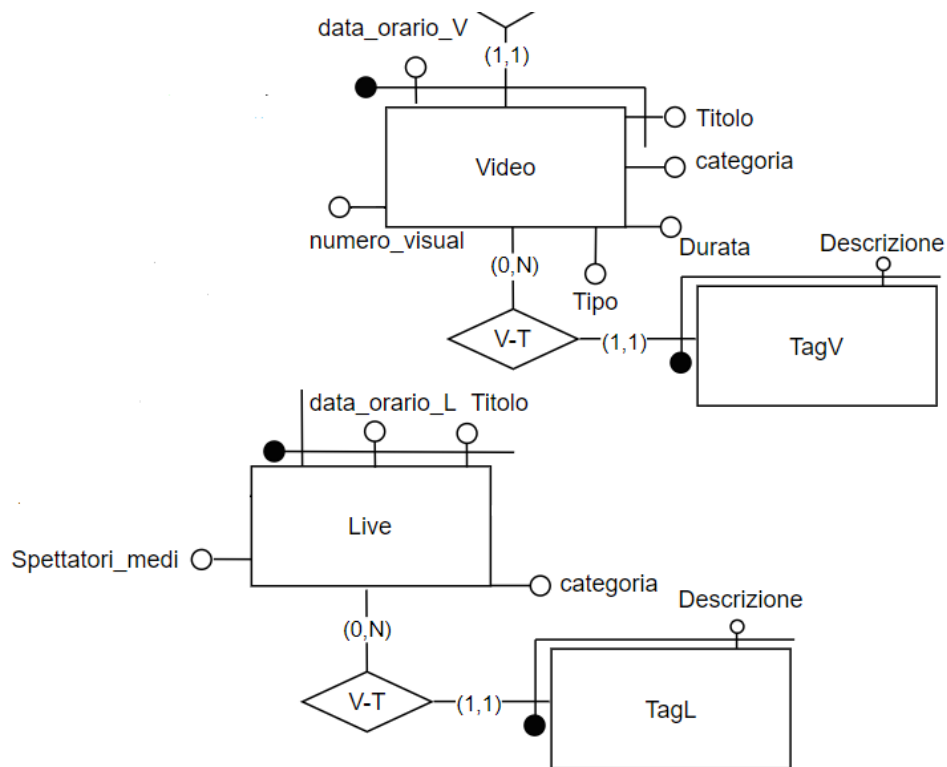
caso 1



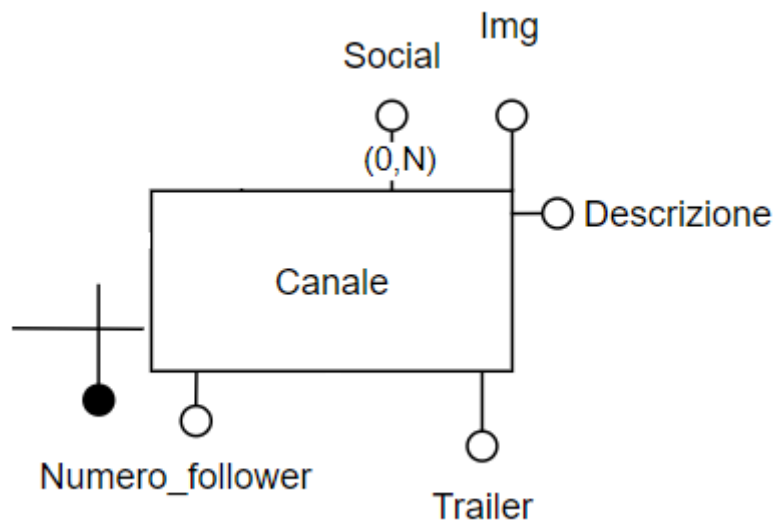
essendo **Tag** un attributo di un'entità che è stata eliminata durante il processo di **eliminazione delle generalizzazioni** dovremmo aggiungere due entità:

- **TagL** (tag live) relazionata all'entità **Live**
- **TagV** (tag video) relazionata all'entità **Video**

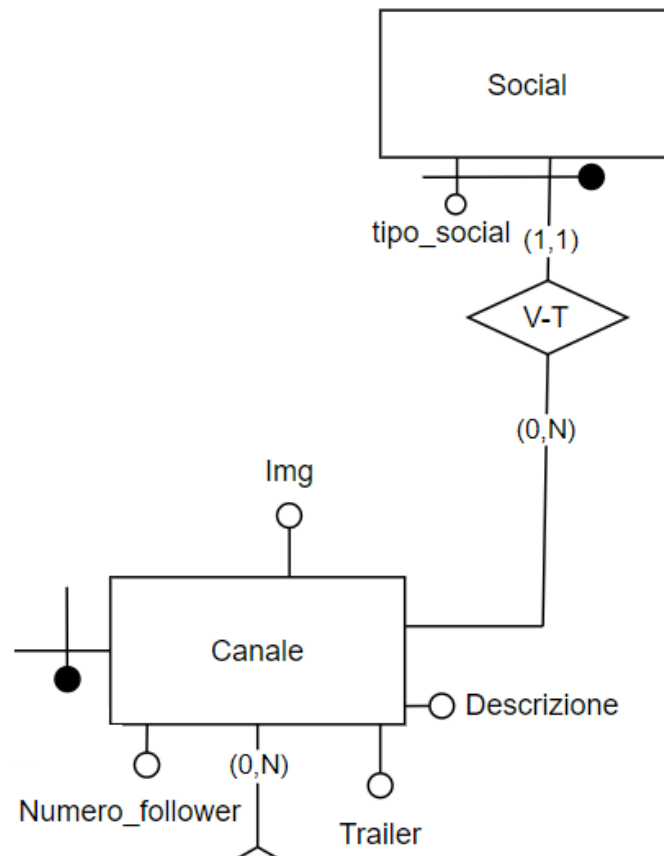
In entrambi i casi andremo ad aggiungere un attributo descrizione e una chiave esterna riferita a **data_orario** delle rispettive entità



caso 2



questo caso è molto simile al precedente, in questo è più semplice e immediato siccome non vi sono generalizzazioni da eliminare e l'attributo **Social** appartiene esclusivamente a **Canale**



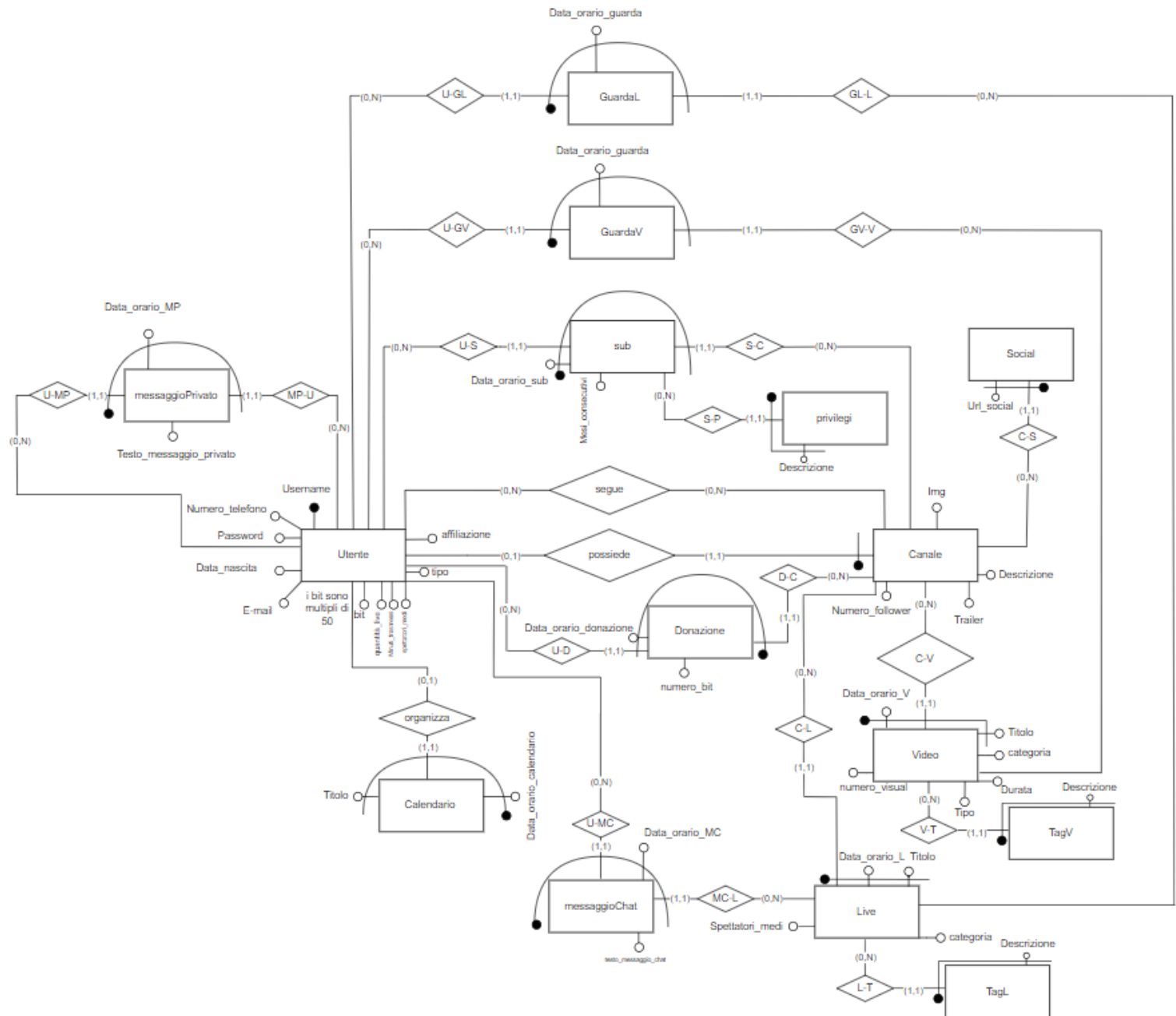
2.3.5 Eventuale scelta degli identificatori principali

Utente	(Username) Username è il modo più semplice per identificare un Utente , infatti ognuno dovrà sceglierne uno differente da qualsiasi altro già precedentemente registrato nel DB
GuardaL	(Data orario guarda, Utente, Live) Rappresenta l'azione che svolge l'utente di guardare una determinata live, è importante che quindi questa entità, generata dalla sostituzione della relazione di guardare una live, che sarebbe stata (0,N) , dovrà essere identificata dalle chiavi primarie di Utente e di Canale , con l'aggiunta di un Orario , che permette di memorizzare le varie azioni giornaliere svolte dallo stesso Utente verso la medesima Live
GuardaV	(Data orario guarda, Utente, Video) Rappresenta l'azione che svolge l'utente di guardare un determinato video o clip, a seconda del tipo di video, è importante che quindi questa entità, generata dalla sostituzione della relazione di guardare un video, che sarebbe stata (0,N) , dovrà essere identificata dalle chiavi primarie di Utente e di Canale , con l'aggiunta di un Orario , che permette di memorizzare le varie azioni giornaliere svolte dallo stesso Utente verso il medesimo Video

sub	<p>(Data orario sub, Utente, Canale)</p> <p>E' l'entità che memorizza tutte le sub che vengono svolte da un Utente, è importante quindi utilizzare come chiave l'identificatore di Utente e di Canale, con l'aggiunta della Data e dell'orario in modo da consentire all'utente di riabbonarsi a quel canale, va inserito un vincolo in cui la sub scade dopo un mese e solo in quel momento è possibile rinnovarla</p>
privilegi	<p>(Descrizione, sub)</p> <p>Serve a memorizzare tutti i privilegi che ottiene l'utente quando fa la sub ad un determinato canale, per questo motivo come chiave primaria avremmo bisogno di sub, che viene identificata, come visto in precedenza utilizzando sia utente che canale, descrizione la aggiungiamo per descrivere il tipo di privilegio, viene aggiunta alla chiave primaria per permettere ad una sub di avere più di un privilegio, tutti differenti.</p>
Social	<p>(Url social, Canale)</p> <p>Serve per dare la possibilità ad un canale di possedere più di un social (massimo 5), per far ciò abbiamo bisogno di identificare il proprietario del canale e aggiungerci il link del social, in questo modo l'entità Social verrà identificato da un canale esistente all'interno del DB ed un social a piacimento inserito dall'Utente proprietari del canale</p>
Canale	<p>(Utente)</p> <p>Un canale deve appartenere ad un unico utente, non tutti gli utenti però possiedono un canale, infatti solamente quelli che sono categorizzati come tipo (attributo inserito in utente) streamer avranno un canale da gestire</p>
Video	<p>(data orario V, Titolo, Canale)</p> <p>L'entità video rappresenta tutte le live finite che sono state salvate come Video o Clip a seconda del tipo (attributo presente in Video) per identificarla abbiamo bisogno quindi di 3 attributi che sono data e orario in cui il video è stato caricato, l'utente che possiede il canale e il titolo del video, in questo modo permettiamo ad un Canale di salvare video ogni giorno in ogni momento della giornata, ma non due video duplicati, cioè non è permesso caricare lo stesso video o clip nel medesimo momento con lo stesso titolo</p>
TagV	<p>(Descrizione, Video)</p> <p>E' l'entità che serve ad elencare tutti i tag che possiede un determinato video, prima era un attributo multiplo, è stato quindi trasformato in entità con l'aggiunta a sua volta di un attributo Descrizione, in questo modo ogni video può avere più di un tag, tutti differenti, senza ripetizioni</p>
Live	<p>(data orario L, Titolo, Canale)</p> <p>l'entità Live rappresenta la streaming avviata da un canale, viene identificata in modo praticamente identico a video, anche lei ha bisogno di conoscere e identificare il canale da cui viene trasmessa, la data in cui viene avviata e il titolo, tutto ciò non permette la creazione di duplicati di live</p>
TagL	<p>(Descrizione, Live)</p> <p>E' l'entità che serve ad elencare tutti i tag che possiede una determinata Live, anche in questo caso come in TagV prima era un attributo multiplo</p>

	<p>dell'entità madre di live, bisogna quindi trasformato in entità con l'aggiunta a sua volta di un attributo Descrizione, in questo modo ogni live può avere più di un tag, tutti differenti, senza ripetizioni</p>
Donazione	<p>(Data orario donazione, Canale, Utente) Come procedimento è molto simile se non identico a quello dell' entità sub, anche donazione ha bisogno di identificare l'utente mittente e il canale destinatario, e per permettere ad un utente di donare più volte allo stesso canale è necessario aggiungere alla chiave un altro attributo che in questo caso, come per le sub, è la data e l'orario</p>
Calendario	<p>(Titolo, Data orario calendario) L'entità calendario deve avere come identificatore sicuramente Username di un utente, successivamente dobbiamo dare la possibilità all'utente di aggiungere titoli diversi alla stessa ora e lo stesso titolo in momenti differenti, per rendere la gestione degli orari e degli avvisi più varia possibile, senza incorrere in duplicati.</p>
MessaggioChat	<p>(Utente, Data orario MC, Live) E' l'entità che rappresenta un messaggio inviato durante una live, il messaggio verrà identificato attraverso l'username dell'utente che lo invia, l'identificatore della live in cui è stato inviato e l'identificazione temporale, in questo modo un Utente potrà inviare in una live più messaggi.</p>
MessaggioPrivato	<p>(Utente, Utente, Data orario MP) E' l'entità che rappresenta un messaggio inviato in chat privata tra due utenti, il messaggio verrà identificato attraverso l'username dell'utente che lo invia e di quello che lo riceve , in questo modo un Utente potrà inviare e ricevere più messaggi, la data e l'orario servono a non permettere la duplicazione di un messaggio, per questo va inserita nella chiave.</p>

2.4 Schema E-R ristrutturato + *business rules*



Modello sviluppato con l'utilizzo di Draw.io

https://drive.google.com/file/d/1Q6U5crG9ImZ2-csxUPNoO9nBtRaPJRcy/view?usp=drive_link

Business rule descrittiva

<i>Entità</i>	<i>Descrizione</i>	<i>Attributi</i>	<i>Identificatore</i>
Utente	utenti che usufruiscono dei servizi del sito	Username, numero_telefono, password, data_nascita, e-mail, bit, affiliazione, minuti_trasmessi, spettatori_medi, tipo	Username
Calendario	calendario utilizzato dall'utente con tipo "streamer" per segnare eventi	Data_orario_calendario, titolo	Username, Data_orario_calendario, titolo
Donazione	Donazione di bit che viene fatta al canale	Numero_bit, Data_orario_donazione	Username, Data_orario_donazione
Canale	Il canale posseduto da ogni Utente di tipo "streamer"	numero_follower, social, img, descrizione, trailer	Username
Live	è un contenuto multimediale trasmetto in diretta	Data_orario_L, titolo, categoria, durata, numero_visualizzazioni, spettatori_medi	Data_orario_L, titolo, Username
Video	è un contenuto multimediale che rappresenta una live passata	Data_orario_V, titolo, categoria, durata, numero_visualizzazioni, Tag, numero_visual	Data_orario_V, titolo, Usernamea
Sub	abbonamento di un utente al canale di uno streamer	mesi_consecutivi, data_orario_sub	Data_orario_sub, Username, Username
GuardaL	rappresentazione di un utente che guarda una live	Data_orario_guarda	Data_orario_guarda, Username, titolo, Data_orario_L
GuardaV	rappresentazione di un utente che guarda un video	Data_orario_guarda	Data_orario_guarda, Username, titolo, Data_orario_V
MessaggioPrivato	Messaggio che un utente invia privatamente ad un altro utente	Data_orario_MP	Data_orario_MP, Username, Username
MessaggioChat	Messaggio che un utente invia nella chat di una live	Data_orario_MC	Data_orario_MP, Username, Data_orario_CM, Titolo,

TagV	Tag appartenenti ad un video	Descrizione	Descrizione, Data_orario_V, Titolo, Username
TagL	Tag appartenenti ad una live	Descrizione	Descrizione, Data_orario_L, Titolo, Username
privilegi	Privilegi ottenuti da una sub ad un canale	Descrizione	Descrizione, Username, Username, Data_orario_sub
Social	Elenco dei social di un canale	Url_social	Url_social, Username

Tabella delle relazioni

Relazione	Descrizione	Entità	Attributi
segue	Associa degli utenti a dei canali	Utente (0,N) Canale (0,N)	
possiede	Associa un utente (streamer) ad un canale	Utente(0,1) Canale(1,1)	
U-S	Associa un utente ad una sub effettuata	Utente (0,N) Sub (1,1)	
S-C	Associa una sub ad un canale	Sub (1,1) Canale (0,N)	
U-GV	Associa utente a GuardaV (guarda video)	Utente (0,N) GuardaV (1,1)	
U-GL	Associa utente a GuardaL (guarda live)	Utente(0,N) GuardaL (1,1)	
GL-Live	Associa il GuardaL all'entità Live	GuardaL (1,1) Live (0,N)	
GV-Video	Associa il GuardaV all'entità Video	GuardaV (1,1) Video (0,N)	
C-V	Associa un canale ad un video	Canale (0,N) Video (1,1)	
C-L	Associa un canale ad una live	Canale (0,N) Live (1,1)	
U-MC	Associa utente a messaggio in chat	Utente (0,N) messaggioChat (1,1)	

MC-L	Associa messaggio in chat ad una live	messagingChat (1,1) Live (0,N)	
organizza	Associa un utente (streamer) ad un calendario	Utente(0,1) Calendario (1,1)	
D-C	Associa una donazione ad un canale	Donazione (1,1) Canale (0,N)	
U-D	Associa un utente ad una donazione	Utente (0,N) Donazione (1,1)	
U-MP	Associa utente a messaggio privato	Utente (0,N) messaggioPrivato (1,1)	
MP-U	Associa messaggio privato ad un utente	messagingPrivato (1,1) Utente (0,N)	
S-P	Associa una sub a dei privilegi	sub (0,N) privilegi (1,1)	
C-S	Associa un canale a dei social	Canale (0,N) Social (1,1)	
V-T	Associa un video a dei tag	Video (0,N) TagL (1,1)	
L-T	Associa una live a dei tag	Live (0,N) TagL (1,1)	

Business rule non descrittiva (schema ristrutturato)

Tutte le regole di vincoli e derivazioni precedentemente segnate con il modello E-R non ristrutturato **valgono ancora**, quelle segnate in questo paragrafo sono regole che si **aggiungono** alle precedenti

- L'entità **Utente** ha un attributo booleano "**tipo**", se ha valore "**true**" l'utente sarà anche uno **streamer**, il valore di default è **false**
- Solo un **Utente** con tipo "**true**" (streamer) ha un **canale** e gli attributi: **quantità_live**, **Minuti_trasmessi**, **spettatori_medi** e **affiliato**
- Un **Utente** di tipo "**streamer**" **non può** effettuare un **abbonamento** al suo stesso **canale**
- Un **Utente non registrato** potrà comunque guardare **live video** e **clip passivamente**
- La data e l'orario di un "**messagingChat**" dovrà essere successiva alla data e orario dell'inizio della **live**
- L'entità video possiede un attributo tipo, se settato a "**true**" il video sarà anche una **clip**, il valore di default è **false**
- Un **Utente (non streamer)** può **guardare live, video** e **clip**, può **mandare messaggi in chat** durante le live e ad altri utenti in **chat privata**, che potrà a sua volta **rispondere** al tuo **messaging** e può **seguire un canale** e **abbonarsi** ricevendo dei **privilegi**.
- Un **Canale** non può essere **relazionato** a più di **5 social**.

2.5 Schema relazionale:

Utente (Username, Password, E-mail*, Data_nascita, Numero_telefono*, tipo, affiliazione, quantità_live*, spettatori_medi*, bit, minuti_trasmessi*)

Canale (Utente, Img*, Trailer*, Descrizione*)

Social (Url_social, Canale)

Sub (Utente, Canale, Data_orario_inizio, Mesi_consecutivi)

Privilegi (Canale, Utente, Data_orario_sub, Descrizione)

Donazione (Utente, Canale, Data_orario_donazione, numero_bit)

Segue (Utente, Canale)

Live (Canale, Data_orario_L, Titolo, categoria, Spettatori_medi)

Video (Canale, Titolo, Data_orario_V, Durata, Tipo, categoria, numero_visual)

TagL (Canale, Data_orario_L, titolo, Descrizione)

TagV (Canale, Titolo, Data_orario_V, Descrizione)

GuardaL (Utente, Data_orario_guarda, Canale, Data_orario_L, Titolo)

GuardaV (Utente, Data_orario_guarda, Canale, Data_orario_L, Titolo)

Calendario (Utente, Data_orario_calendario, Titolo)

MessaggioChat (Utente, Canale, Data_orario_L, Data_orario_MC, Titolo, testo_messaggio_chat)

MessaggioPrivato (Utente_mittente, Utente_destinatario, data_orario_MP, Testo_messaggio_privato)

Vincoli di integrità referenziale:

Canale(Utente) referencia **Utente(Username)**

Social(Canale) referencia **Canale(Utente)**

Follow(Canale) referencia **Canale(Utente)**

Subscriptio(Canale) referencia **Canale(Utente)**

Sub(Utente) referencia **Utente(Username)**

Donazione(Utente) referencia **Utente(Username)**

Donazione(Canale) referencia **Canale(Utente)**

Live(Canale) referencia **Canale(Utente)**

Video(Canale) referencia **Canale(Utente)**

Privilegi(Canale) referencia **Canale(Utente)**

Privilegi(Data_orario_sub) referencia **Subscription(Data_orario_sub)**

Privilegi(Utente) referencia **Utente(Username)**

Calendario(Utente) referencia **Utente(Username)**

TagL(Canale) referencia **Canale(Utente)**

TagL(Data_orario_L) referencia **Live(Data_orario_L)**

TagV(Titolo) referencia **Video(Titolo)**

TagV(Canale) referencia **Canale(Utente)**

TagV(Data_orario_V) referencia **Video(Data_orario_V)**

messaggioChat(Utente) referencia **Utente(Username)**

messaggioChat(Canale) referencia **Canale(Utente)**

messaggioChat(DataOraInizioLive) referencia **Live(DataOraLive)**

messaggioChat(titololoLive) referencia **Live(Titolo)**

messaggioPrivato(UtenteMittente) referencia **Utente(Username)**

messaggioPrivato(UtenteDestinatario) referencia **Utente(Username)**

GuardaV (Utente) referencia **Utente(Username)**

GuardaV(Canale) referencia **Canale(Utente)**

GuardaV(Data_orario_V) referencia **Video(Data_orario_V)**

GuardaL(Utente) referencia **Utente(Username)**

GuardaL(Canale) referencia **Canale(Utente)**

GuardaL(Data_orario_L) referencia **Live(Data_orario_L)**

3. Implementazione

3.1 DDL di creazione del database

```
CREATE TABLE Utente(  
  Username VARCHAR(20) NOT NULL UNIQUE,  
  numero_telefono VARCHAR(15),  
  Password VARCHAR(30) NOT NULL,  
  data_nascita DATE NOT NULL,  
  email VARCHAR(20),  
  bit INTEGER DEFAULT 0,  
  quantita_live INTEGER CHECK(tipo) DEFAULT NULL,  
  minuti_trasmessi INTEGER CHECK(tipo) DEFAULT NULL,  
  spettatori_medi INTEGER CHECK(tipo) DEFAULT NULL,  
  affiliazione boolean default false,  
  tipo boolean default false,  
  PRIMARY KEY (Username)  
);  
  
CREATE TABLE Canale(  
  username VARCHAR(20),  
  IMAGE VARCHAR(500),  
  Descrizione TEXT,  
  Trailer VARCHAR(500),  
  PRIMARY KEY (username),  
  FOREIGN KEY (username) REFERENCES utente(username)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
);  
  
CREATE TABLE calendario(  
  username VARCHAR(20),  
  Titolo VARCHAR(30) NOT NULL,  
  Data_orario_calendario TIMESTAMP NOT NULL UNIQUE,  
  PRIMARY KEY (Username, Data_orario_calendario, Titolo),  
  FOREIGN KEY (Username) REFERENCES utente(username)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
);
```

```

CREATE TABLE messaggioPrivato(
username1 VARCHAR(20),
username2 VARCHAR(20),
testo_messaggio_privato TEXT NOT NULL,
Data_orario_MP TIMESTAMP NOT NULL,
PRIMARY KEY (username1,username2, Data_orario_MP),
FOREIGN KEY(username1) REFERENCES utente(username)
    ON UPDATE CASCADE
    ON DELETE SET NULL,
FOREIGN KEY(username2) REFERENCES utente(username)
    ON UPDATE CASCADE
    ON DELETE SET NULL
);

CREATE TABLE Live(
username VARCHAR(20),
Titolo VARCHAR(30) NOT NULL,
data_orario_L TIMESTAMP NOT NULL,
spettatori_medi INTEGER NOT NULL DEFAULT 0,
Categoria VARCHAR(20) NOT NULL DEFAULT 'JustChatting',
PRIMARY KEY (username, Titolo,data_orario_L),
FOREIGN KEY (username) REFERENCES canale(username)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

```

```

CREATE TABLE sub(
usernameutente VARCHAR(20),
data_orario_sub TIMESTAMP NOT NULL,
mesi_consecutivi INTEGER NOT NULL,
usernameCanale VARCHAR(20),
PRIMARY KEY (usernameutente,data_orario_sub, usernamecanale),
FOREIGN KEY(usernameutente) REFERENCES utente(username)
    ON UPDATE CASCADE
    ON DELETE SET NULL,
FOREIGN KEY(usernamecanale) REFERENCES canale(username)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

```

```

CREATE TABLE donazione(
usernameutente VARCHAR(20),
data_orario_donazione TIMESTAMP NOT NULL,
numero_bit INTEGER NOT NULL,
usernamecanale VARCHAR(20),
PRIMARY KEY (usernameutente,data_orario_donazione, usernamecanale),
FOREIGN KEY(usernameutente) REFERENCES utente(username)
    ON UPDATE CASCADE
    ON DELETE SET NULL,
FOREIGN KEY(usernamecanale) REFERENCES canale(username)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

```

```

CREATE TABLE messaggioChat(
username VARCHAR(20),
data_orario_mc TIMESTAMP NOT NULL,
testo_messaggio_chat TEXT NOT NULL,
data_orario_L TIMESTAMP,
titolo VARCHAR(30),
usernamelive VARCHAR(20),
PRIMARY KEY (username,data_orario_MC, data_orario_L, titolo,usernamelive),
FOREIGN KEY(username) REFERENCES utente(username)
    ON UPDATE CASCADE
    ON DELETE SET NULL,
FOREIGN KEY(usernamelive, data_orario_L,titolo) REFERENCES live(username,data_orario_L,titolo)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

```



```

CREATE TABLE guardaV(
usernameutente VARCHAR(20),
data_orario_guarda TIMESTAMP NOT NULL,
data_orario_v TIMESTAMP,
titolo VARCHAR(30),
usernamecanale VARCHAR(20),
PRIMARY KEY (usernameutente,data_orario_guarda, data_orario_v,titolo,usernamecanale),
FOREIGN KEY(usernameutente) REFERENCES utente(username)
    ON UPDATE CASCADE
    ON DELETE SET NULL,
FOREIGN KEY(usernamecanale, data_orario_v,titolo) REFERENCES video(usernamecanale,data_orario_v,titolo)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

CREATE TABLE guardaL(
usernameutente VARCHAR(20),
data_orario_guarda TIMESTAMP NOT NULL,
data_orario_L TIMESTAMP,
titolo VARCHAR(30),
usernamelive VARCHAR(20),
PRIMARY KEY (usernameutente,data_orario_guarda, data_orario_L, titolo,usernamelive),
FOREIGN KEY(usernameutente) REFERENCES utente(username)
    ON UPDATE CASCADE
    ON DELETE SET NULL,
FOREIGN KEY(usernamelive, data_orario_L,titolo) REFERENCES live(username,data_orario_L,titolo)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

CREATE TABLE video(
usernamecanale VARCHAR(20),
data_orario_v TIMESTAMP NOT NULL,
titolo VARCHAR(30) NOT NULL,
categoria VARCHAR(30) NOT NULL,
durata TIME NOT NULL,
tipo boolean NOT NULL,
numero_visual INTEGER NOT NULL DEFAULT 0,
PRIMARY KEY (usernamecanale,data_orario_v,titolo),
FOREIGN KEY(usernamecanale) REFERENCES canale(username)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

```

```

CREATE TABLE social(
usernamecanale VARCHAR(20),
tipo_social VARCHAR(100),
PRIMARY KEY (usernamecanale, tipo_social),
FOREIGN KEY(usernamecanale) REFERENCES canale(username)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

```

```

CREATE TABLE privilegi(
usernameutente VARCHAR(20),
data_orario_sub TIMESTAMP,
descrizione VARCHAR(20),
usernamecanale VARCHAR(20),
PRIMARY KEY (descrizione, data_orario_sub, usernameutente, usernamecanale),
FOREIGN KEY(usernamecanale, usernameutente, data_orario_sub) REFERENCES sub(usernamecanale, usernameutente, data_orario_sub)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

```

```

CREATE TABLE Segue(
    usernameutente VARCHAR(20),
    usernamecanale VARCHAR(20),
    PRIMARY KEY(usernameutente, usernamecanale),
    FOREIGN KEY(usernameutente) REFERENCES utente(username)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY(usernamecanale) REFERENCES canale(username)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

```



```

CREATE TABLE tagL(
    username VARCHAR(20),
    data_orario_l TIMESTAMP NOT NULL,
    titolo VARCHAR(30),
    descrizione VARCHAR(20),
    PRIMARY KEY(username,data_orario_l,titolo,descrizione),
    FOREIGN KEY(titolo,data_orario_l,username) REFERENCES live(titolo,data_orario_l,username)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

CREATE TABLE tagV(
    usernamecanale VARCHAR(20),
    data_orario_v TIMESTAMP NOT NULL,
    titolo VARCHAR(30) NOT NULL,
    descrizione VARCHAR(20) NOT NULL,
    PRIMARY KEY(descrizione,usernamecanale,data_orario_v,titolo),
    FOREIGN KEY(data_orario_v,titolo,usernamecanale) REFERENCES video(data_orario_v,titolo,usernamecanale)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

```

3.2 DML di popolamento

```

INSERT INTO utente (username,numero_telefono,password,data_nascita,email,bit,quantita_live,
                    minuti_trasmessi,spettatori_medi,affiliazione,tipo)
VALUES('cascett','+393881593805','TwitchABC1234','2002-08-05','mcascione02@gmail.com',100,
      20,1200,300,true,true);

INSERT INTO utente (username,numero_telefono,password,data_nascita,email)
VALUES('pier2690__','+393925399431','LuigiABC1234','2002-06-06','pierbosca@gmail.com');

INSERT INTO utente (username,numero_telefono,password,data_nascita,email)
VALUES('roberto406','+393325339111','Robix04@','2002-09-04','robix04@gmail.com');

INSERT INTO utente (username,numero_telefono,password,data_nascita,email)
VALUES('giups02','+393426312409','Giups02ABC1234','2002-06-09','giups02@gmail.com');

INSERT INTO utente (username,numero_telefono,password,data_nascita,email)
VALUES('fede01','+393945090569','federicaABC01@','2001-03-30','federica01@gmail.com');

INSERT INTO canale(username,image,descrizione,trailer)
VALUES('cascett','https://imageurlPhotoChannelCascett.jpg/','Principalmente Stream di R6, Destiny e Minecraft',
      'https://youtube.com/trailerCascettChanell/');

INSERT INTO calendario(username,titolo,data_orario_calendario)
VALUES('cascett','Stream di R6','2023-07-14 12:34:00');

INSERT INTO messaggioprivato(username1,username2,testo_messaggio_privato,data_orario_mp)
VALUES('fede01','giups02','Guardiamo la live di cascett?','2023-07-14 13:00:00');

INSERT INTO live(username,titolo,data_orario_l,spettatori_medi,categoria)
VALUES ('cascett','Stream di R6','2023-07-14 12:34:00', 200, 'Rainbow Six Siege')
INSERT INTO sub(usernameutente,data_orario_sub,mesi_consecutivi,usernamecanale)
VALUES ('fede01','2023-07-14 12:55:00',3,'cascett')

```

```

INSERT INTO donazione(usernameutente,data_orario_donazione,numero_bit,usernamecanale)
VALUES ('pier2690__','2023-07-14 13:20:00',150,'cascett')
INSERT INTO messaggiochat(username,data_orario_mc,
                           testo_messaggio_chat,data_orario_l,titolo,usernamelive)
VALUES ('pier2690__','2023-07-14 12:50:00','Che rank sei?',
        '2023-07-14 12:34:00','Stream di R6', 'cascett')

INSERT INTO guardal(usernameutente,data_orario_guarda,
                    data_orario_l,titolo,usernamelive)
VALUES('fede01','2023-07-14 12:45:00','2023-07-14 12:34:00',
        'Stream di R6','cascett')

INSERT INTO video(usernamecanale,data_orario_v,titolo,categoria,
                  durata,tipo,numero_visual)
VALUES ('cascett','2023-07-14 12:55:35','1v5 ACE','Gaming, Italiano, Tattico',
        '00:02:40',true,97)
INSERT INTO social(usernamecanale,tiposocial)
VALUES('cascett','https://www.instagram.com/michele_cascione2002/')
INSERT INTO guardav(usernameutente,data_orario_guarda,data_orario_v,
                    titolo,usernamecanale)
VALUES('roberto406','2023-07-19 20:08:35',
        '2023-07-14 12:55:35','1v5 ACE','cascett')

INSERT INTO segue(usernameutente,usernamecanale)
VALUES('pier2690__','cascett')
INSERT INTO tagl(username,data_orario_l,titolo,descrizione)
VALUES('cascett','2023-07-14 12:34:00','Stream di R6','Multiplayer');

INSERT INTO tagl(username,data_orario_l,titolo,descrizione)
VALUES('cascett','2023-07-14 12:34:00','Stream di R6','Sparatutto');

INSERT INTO tagl(username,data_orario_l,titolo,descrizione)
VALUES('cascett','2023-07-14 12:34:00','Stream di R6','Ita');

INSERT INTO tagv(usernamecanale,data_orario_v,titolo,descrizione)
VALUES('cascett','2023-07-14 12:55:35','1v5 ACE','Multiplayer');

INSERT INTO tagv(usernamecanale,data_orario_v,titolo,descrizione)
VALUES('cascett','2023-07-14 12:55:35','1v5 ACE','R6');

```


3.3 Qualche operazione di cancellazione e modifica per verificare i vincoli e gli effetti causati da operazioni su chiavi esterne.

- Operazione di modifica sul titolo di un video: titolo rappresenta una chiave esterna ed è referenziato da guarda_v e tag_v. La modifica del titolo modifica a sua volta tutti i campi delle righe che lo referenziano.

```
UPDATE video
SET titolo = '1v5 COMP'
WHERE titolo = '1v5 ACE';
```

GuardaV:

	usernameutente [PK] character varying (20)	data_orario_guarda [PK] timestamp without time zone	data_orario_v [PK] timestamp without time zone	titolo [PK] character varying (30)	usernamecanale [PK] character varying (20)
1	roberto406	2023-07-19 20:08:35	2023-07-14 12:55:35	1v5 COMP	cascett

TagV:

	usernamecanale [PK] character varying (20)	data_orario_v [PK] timestamp without time zone	titolo [PK] character varying (30)	descrizione [PK] character varying (20)
1	cascett	2023-07-14 12:55:35	1v5 COMP	Multiplayer
2	cascett	2023-07-14 12:55:35	1v5 COMP	R6

- Operazione di modifica dello username di un utente che ha un canale: ogni entità è riconducibile allo username dell'utente. Vengono riportate solo le entità più importanti sulle quali l'update ha effetto.

```
UPDATE utente
SET username = 'pierluigi0602'
WHERE username = 'cascett';
```

Live:

	username [PK] character varying (20)	titolo [PK] character varying (30)	data_orario_l [PK] timestamp without time zone	spettatori_medi integer	categoria character varying (20)
1	pierluigi0602	Stream di R6	2023-07-14 12:34:00	200	Rainbow Six Siege

Segue:

	usernameutente [PK] character varying (20)	usernamecanale [PK] character varying (20)
1	pier2690__	pierluigi0602

Canale:

	username [PK] character varying (20)	image character varying (500)	descrizione text	trailer character varying (500)
1	pierluigi0602	https://imageurlPhotoChannelCascett.jpg/	Principalmente Stream di R6, Destiny e Minecraft	https://youtube.com/trailerCascettChanell/

GuardaL:

	usernameutente [PK] character varying (20)	data_orario_guarda [PK] timestamp without time zone	data_orario_l [PK] timestamp without time zone	titolo [PK] character varying (30)	usernamelive [PK] character varying (20)
1	fede01	2023-07-14 12:45:00	2023-07-14 12:34:00	Stream di R6	pierluigi0602

MessaggioChat:

	username [PK] character varying (20)	data_orario_mc [PK] timestamp without time zone	testo_messaggio_chat text	data_orario_l [PK] timestamp without time zone	titolo [PK] character varying (30)	usernamelive [PK] character varying (20)
1	pier2690__	2023-07-14 12:50:00	Che rank sei?	2023-07-14 12:34:00	Stream di R6	pierluigi0602

- Eliminazione di un video: i tag riferiti a quel video si cancellano a loro volta

```
DELETE FROM video  
WHERE titolo='lv5 COMP'
```

usernamecanale [PK] character varying (20)	data_orario_v [PK] timestamp without time zone	titolo [PK] character varying (30)	descrizione [PK] character varying (20)
---	---	---------------------------------------	--

- Eliminazione di un utente che è uno streamer:

```
DELETE FROM utente  
where username='pierluigi0602'
```

vengono eliminate le entità che sono in relazione con quell'utente. Di conseguenza vengono eliminati elementi come il canale e tutte le relazioni che gli utenti avevano con quel canale come, ad esempio, le sub e i follow.

Segue:

usernameutente [PK] character varying (20)	usernamecanale [PK] character varying (20)
---	---

Tutte le righe coinvolte nella cancellazione dello stream risultano quindi eliminate.