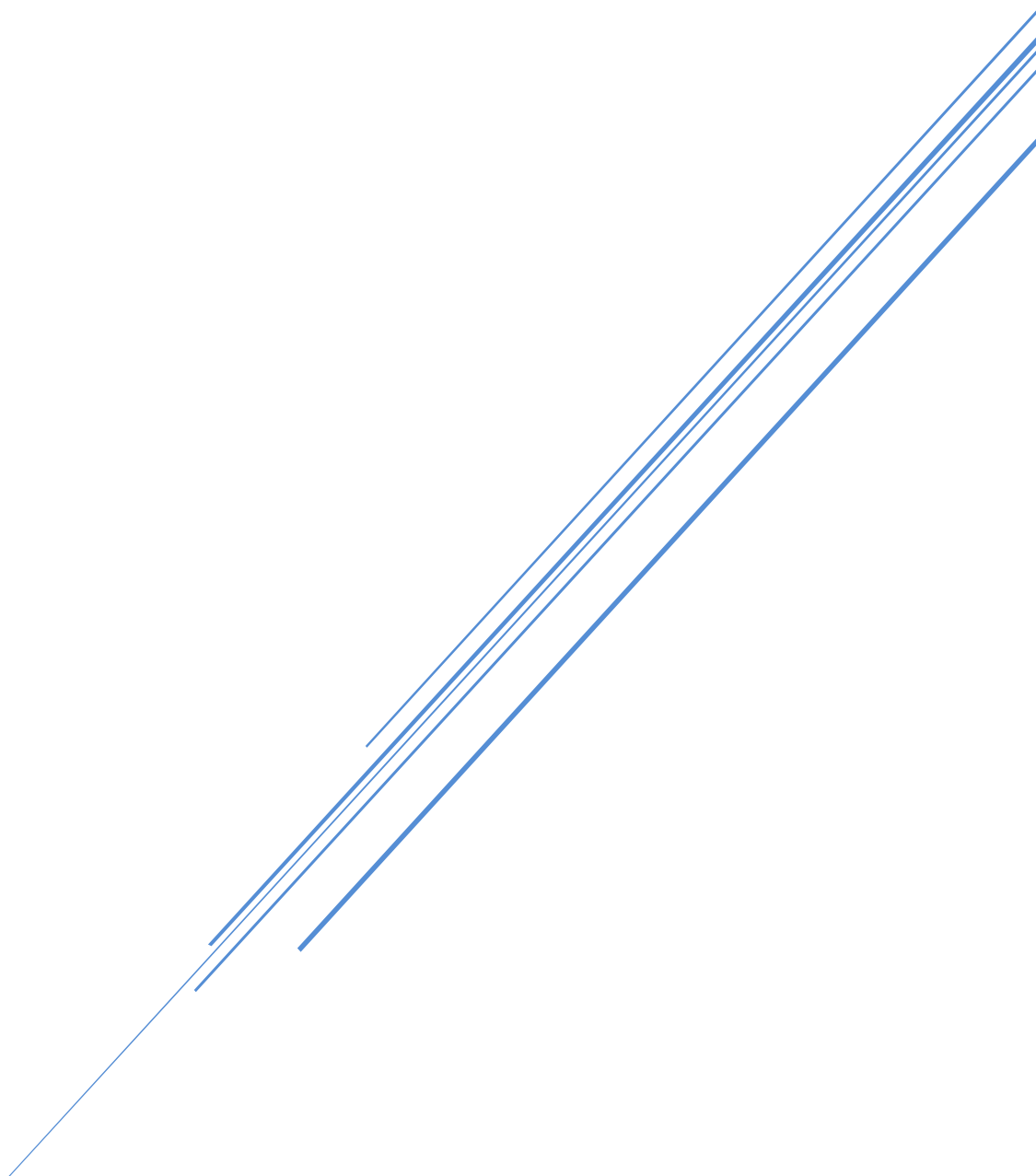


LABORATORIO BASI DI DATI

2022/2023

“Progetto di piattaforma di live streaming”



Sommario

1. Progettazione concettuale	2
1.1 Requisiti iniziali	2
1.2 Glossario dei termini.....	4
1.3 Requisiti rivisti	5
1.3.1 Gruppi di frasi omogenee	8
1.4 Schema ER + Business rules	10
1.4.1 Schema ER	10
1.4.2 Business rules	11
2.0 Progettazione Logica	12
2.1 Tavola dei volumi.....	12
2.2 Tavola delle operazioni.....	13
2.3 Ristrutturazione dello schema ER.....	15
2.3.1 Analisi delle ridondanze.....	15
2.3.2 Eliminazione delle generalizzazioni	19
2.3.3 Eventuale eliminazione degli attributi composti e degli attributi multivalore.....	23
2.4 Schema E-R ristrutturato + business rules.....	25
2.5 Schema relazionale	27
3.0 Implementazione SQL.....	29
3.1 DDL di creazione del database.....	29
3.2 DML di popolamento di tutte le tabelle del database.....	32
3.3 Qualche operazione di cancellazione e modifica per verificare i vincoli e gli effetti causati da operazioni su chiavi esterne	33

1. Progettazione concettuale

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. 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 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.

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 follower 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.

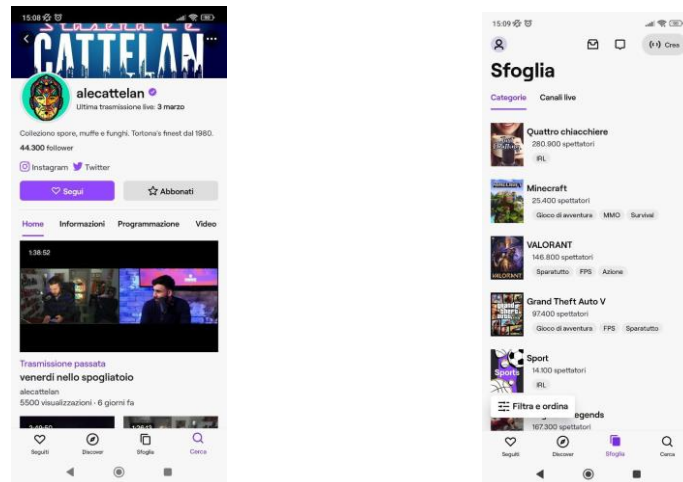


Figura 1 (a) La home del canale di alecattelan. (b) La lista delle categorie

¹ Il servizio descritto è liberamente ispirato a Twitch (<https://www.twitch.tv>) a cui è possibile fare riferimento per completare e disambiguare i requisiti.

1.2 Glossario dei termini

Termine	Descrizione	Sinonimo	Collegamenti
Streamer	<i>Utente che trasmette i contenuti multimediale sulla piattaforma</i>	<i>Utente</i>	<i>Canale, Affiliato, sub, bit, live</i>
Viewer (Spettatore)	<i>Utente che accede alla piattaforma per visualizzare i contenuti messi a disposizione dagli streamer.</i>	<i>Utente</i>	<i>Canale, bit, sub, live</i>
Live	<i>Rappresenta una live (Trasmissione in diretta) trasmessa da uno streamer</i>	<i>Trasmissione in diretta</i>	<i>Streamer, canale</i>
Clip	<i>Rappresenta una clip di breve durata associata a una live passata</i>	<i>Video breve</i>	<i>Streamer, canale</i>
Affiliato	<i>Rappresenta uno streamer che rispetta determinati parametri di performance.</i>	<i>/</i>	<i>Streamer, canale</i>
Sub	<i>Abbonamento a pagamento al canale di uno streamer che offre ai viewer dei privilegi (emoticon personalizzate, ecc.).</i>	<i>Abbonamento</i>	<i>Viewer, Streamer, live</i>
Bit	<i>Moneta virtuale che può essere acquistata dagli utenti sulla piattaforma e utilizzata per fare donazioni agli streamer</i>	<i>Moneta virtuale</i>	<i>Viewer, Streamer, live</i>
Canale	<i>Rappresenta un canale di uno streamer</i>	<i>/</i>	<i>Streamer, live</i>
Messaggio (ChatPrivata-ChatLive)	<i>Strumento attraverso il quale gli utenti possono comunicare in modo sincrono durante le live o in modo asincrono tramite messaggi privati.</i>	<i>Messaggio</i>	<i>Streamer, Canale, Live</i>
Video	<i>Registrazione di un live passato</i>	<i>/</i>	<i>Streamer, Canale</i>

Calendario	<i>Strumento utilizzato dagli streamer per indicare quando faranno la prossima live.</i>	<i>Programma live</i>	<i>Streamer</i>
Follow	<i>Un follower su Twitch è una persona che ha scelto di seguire un canale specifico</i>	/	<i>Viewer, canale</i>

1.3 Requisiti rivisti

Si vuole realizzare una base di dati per un servizio che permette di fare live streaming su vari argomenti¹.

~~Il live streaming (o, più sinteticamente, le live)~~ permettono di interagire con il pubblico in tempo reale grazie a feed video, chat e altro.

Ogni L' utente può essere rappresentato come ~~può essere~~ spettatore o streamer, o entrambi. Gli spettatori possono essere registrati ~~al servizio~~ oppure possono guardare le live in modo anonimo. Per gli iscritti rappresentiamo ~~per registrarsi, gli utenti devono indicare~~ nome utente, password, data di nascita, numero di telefono o indirizzo mail. ~~Gli utenti iscritti~~ Gli iscritti possono chattare, seguire lo streamer, creare dirette.

Lo streamer ~~hanno ciascuno~~ possiede un canale. ~~che può essere caratterizzato tramite~~ Il canale può possedere una descrizione. Per il canale rappresentiamo ~~Per ogni canale, è possibile specificare, un nome canale(nickname),~~ 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. Per le live rappresentiamo ~~Ognuno ha~~ un titolo, una durata, ~~appartiene a~~ una categoria (Figura 1(b)) e ~~eventuali 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.

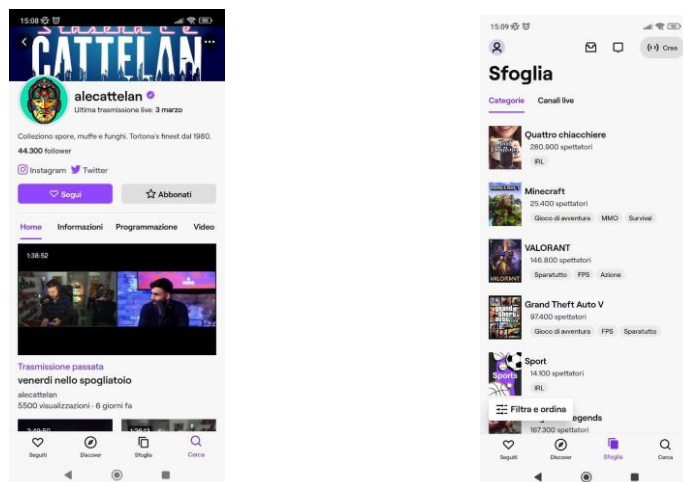


Figura 1 (a) La home del canale di alecattelan. (b) La lista delle categorie

¹ Il servizio descritto è liberamente ispirato a Twitch (<https://www.twitch.tv>) a cui è possibile fare riferimento per completare e disambiguare i requisiti.

Per gli streamer ~~Per ogni creatore di contenuti~~, ~~rappresentiamo~~ ~~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*. Per le stream ~~hanno degli~~ ~~rappresentiamo~~ degli orari. Per gli ~~Ogni~~ streamer ~~inoltre rappresentiamo~~ ~~ha~~ un calendario in cui ~~può~~ ~~possono~~ dire quando ~~farà~~ ~~faranno~~ stream e indicare il titolo delle prossime live.

~~I viewer~~ Gli spettatori, possono diventare follower del canale degli streamer che preferiscono, e le loro preferenze sono raccolte in un elenco di follower a cui possono accedere dal loro profilo. ~~I viewer~~ Gli spettatori possono inoltre supportare gli streamer tramite la subscription (a pagamento) al loro canale, ottenendo dei privilegi (emoticon personalizzate, ecc.). Inoltre, ~~gli utenti~~ ~~gli spettatori~~ 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~~ ~~gli iscritti~~ 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.

1.3.1 Gruppi di frasi omogenee

➤ Frasi relative agli utenti:

- *L'utente può essere rappresentato come può essere spettatore o streamer, o entrambi.*
- *Gli iscritti rappresentiamo il nome utente, password, data di nascita, numero di telefono o indirizzo mail.*
- *Gli utenti iscritti possono chattare, seguire lo streamer, creare dirette*

➤ Frasi relative agli streamer

- *Per gli streamer, rappresentiamo il canale, 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, inoltre rappresentiamo un calendario in cui possono dire quando faranno stream e indicare il titolo delle prossime live.*
- *Ogni streamer ha un calendario in cui può dire quando farà stream e indicare il titolo delle prossime live.*
- *Quando uno streamer rispetta determinati parametri di performance può diventare affiliate*

➤ Frasi relative agli spettatori

- *Gli spettatori possono essere registrati al servizio oppure possono guardare le live in modo anonimo, inoltre possono diventare follower del canale degli streamer che preferiscono, e le loro preferenze sono raccolte in un elenco di follower a cui possono accedere dal loro profile*
- *I viewer possono inoltre supportare gli streamer tramite la subscription (a pagamento) al loro canale, ottenendo dei privilegi (emoticon personalizzate, ecc.)*

➤ Frasi relative alle Live

- *Per le live rappresentiamo un titolo, una durata, una categoria e tag*
- *Per ogni live viene memorizzato il numero medio di spettatori mentre per i video e le clip il numero di visualizzazioni.*
- *Le live possono anche non diventare video del canale.*

➤ Frasi relative alle clip

- *Video di durata breve presenti sul canale degli streamer*

➤ Frasi relative ai video

- *Ogni video 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 video del canale.*

➤ Frasi relative agli affiliati

- *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.*

➤ Frasi relative alle sub

- *Gli spettatori possono supportare gli streamer tramite la subscription (a pagamento) al loro canale, ottenendo dei privilegi (emoticon personalizzate, ecc.)*

➤ **Frase relative ai bit**

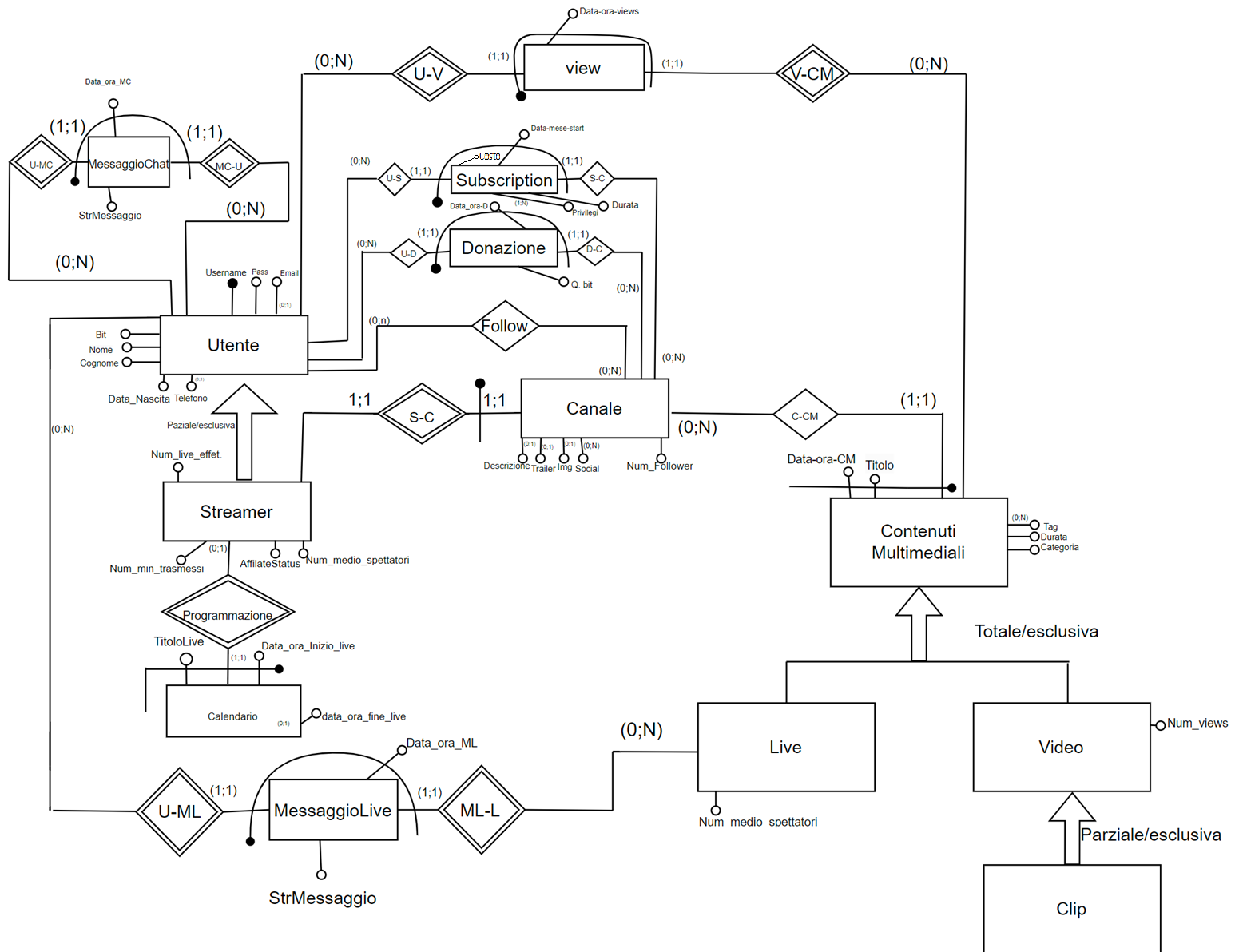
- *Gli spettatori hanno un portafoglio di bit (moneta virtuale che possono acquistare tramite la piattaforma), che possono usare per fare donazioni agli streamer*

➤ **Frase relative ai canali**

- *Per il canale rappresentiamo, una descrizione ,una lista di social associati (ad esempio Instagram, YouTube, ecc.), un'immagine profilo e anche un trailer. Nel canale possono esserci live, video (live passate) e clip (video di durata breve).*

1.4 Schema ER + Business rules

1.4.1 Schema ER



1.4.2 Business rules

1. Solo gli utenti registrati possono chattare, seguire gli streamer, e creare live.
2. Gli spettatori possono accedere alle live in modo anonimo senza registrarsi, ma se decidono di registrarsi devono fornire tutte le informazioni richieste.
3. Gli streamer possono avere un solo canale associato al proprio account e devono fornire una descrizione del canale e tutte le informazioni richieste.
4. Nei canali possono essere presenti live (trasmissione in diretta), video (live passate) e clip.
5. I viewer possono guardare una sola live per volta. (Viewer → Utente) il viewer è l'utente.
6. Lo streamer può avviare una live per volta.
7. L'elenco di follower di un utente deve contenere solo utenti registrati.
8. Gli utenti possono inviare messaggi privati solo ad altri utenti registrati.
9. Il numero di telefono o l'indirizzo e-mail forniti durante la registrazione deve essere univoco per ogni utente iscritto. Se viene inserito il telefono allora l'email non è richiesta e viceversa.
10. Il numero medio di spettatori per ogni live deve essere un valore numerico positivo. (≥ 0)
11. Il numero di visualizzazioni per ogni video o clip deve essere un valore numerico positivo. (≥ 0)
12. Gli utenti possono effettuare donazioni solo se dispongono di fondi sufficienti nel loro portafoglio di bit.
13. Il titolo di una live non può essere vuoto.
14. I proprietari dei canali possono solo modificare ed eliminare i propri canali e i dati associati.
15. Uno streamer può diventare affiliate se rispetta determinati parametri di performance (500 minuti trasmessi, una media di tre o più spettatori simultanei, almeno 50 follower)
16. Il numero di bit donato deve essere > 0
17. $\text{Data_ora_ML} \geq \text{Data_ora_inizio_Live}$
18. Per iscriversi l'utente deve fornire un email valida o un numero di telefono valido
19. Le live possono diventare video e le clip (video brevi) sono estratti di video.
20. Ogni streamer ha un calendario in cui può dire quando farà stream e indicare il titolo delle prossime live. La data_ora di inizio (e fine live), e il titolo, devono corrispondere a quanto riportato sul calendario dello streamer.

2.0 Progettazione Logica

2.1 Tavola dei volumi

Concetto	Tipo	Volume	Motivazione
Utente	E	70 milioni	*
Utente Streamer	E	30 milioni	*
ContenutiMultimediali	E	200 milioni	*
Live	E	120 milioni	*
Video	E	70 milioni	*
Clip	E	60 milioni	*
Canale	E	30 milioni	*
Calendario Streamer	E	25 milioni	*
Subscription	E	15 milioni	*
Donazione	E	40 milioni	*
Follow	E	30 milioni	*
(Utente) MC-U (MessaggioChat)	A	300 milioni	*
(MessaggioChat) U-MC (Utente)	A	300 milioni	*
(Utente) U-ML (MessaggioLive)	A	300 milioni	*
(MessaggioLive) ML-L (Live)	A	120 milioni	*
MessaggioChat	E	30 milioni	*
MesaggioLive	E	500 milioni	*
(Utente) U-V (view)	A	500 milioni	*
(view) V-CM (ContenutiMultimediali)	A	30 milioni	
(Streamer) Programmazione (Calendario)	A	25 milioni	*
(Streamer) S-C (Canale)	A	30 milioni	*
(Canale) C-CM (ContenutiMultimediali)	A	200 milioni	*

N.B. * → Fonte stima volumi presa online : (<https://twitchtracker.com/statistics>) , (<https://www.demandsage.com/twitch-users>) , (<https://earthweb.com/twitch-statistics/>) , (<https://www.streamscheme.com/twitch-statistics/>) , (<https://www.twitch.tv>)

(Valori approssimati NON realistici)

2.2 Tavola delle operazioni

Operazione	Descrizione	Tipo	Frequenza
1. Registrazione	Registrazione di un nuovo utente	Interattiva	10 mila al giorno
2. Creazione live	Creazione di una nuova live da parte dello streamer	Interattiva	500 mila al giorno
3. Creazione video	Creazione di un nuovo video da parte dello streamer	Interattiva	200 mila al giorno
4. Creazione clip	Creazione di una nuova clip da parte dello streamer	Interattiva	90 mila al giorno
5. Chat pubblica	Invio di messaggi nella chat pubblica della live	Interattiva	30 milioni al giorno
6. Chat privata	Invio di messaggi privati tra utenti	Interattiva	1 milioni al giorno
7. Follow	Aggiunta di uno streamer alla lista dei preferiti di un utente	Interattiva	1 milioni al giorno
8. Subscription	Sottoscrizione a pagamento al canale di uno streamer	Interattiva	1 milioni al giorno
9. Donazione	Donazione di bit da parte di un utente a uno streamer	Interattiva	2 milioni al giorno
10. Controllo affiliate	Controllo quotidiano per la qualifica di affiliate	Batch	1 al giorno
11. Classifica degli streamer	Calcolo settimanale della classifica degli streamer più seguiti	Batch	1 a settimana
12. Ottenere il numero di minuti trasmessi di live di uno streamer	Permette il calcolo dei minuti complessivi trasmessi da parte di uno streamer	Batch	1 al giorno
13. Ottenere numero di follower di un determinato canale	Permette di ottenere numero follower di un determinato canale	Batch	1 al giorno
14. Ottenere il numero di donazioni e di Subscription di un determinato canale	Permette di ottenere il numero di donazioni e di Subscription di un determinato canale	Batch	1 a settimana
15. Visualizzazione o ricerca live	Permette agli utenti di visualizzare o ricercare le live disponibili	Interattiva	1 milione al giorno
16. Visualizzazione o ricerca video	Permette agli utenti di visualizzare o di cercare i video salvati nel canale di uno streamer	Interattiva	400 mila al giorno
17. Visualizzazione o ricerca Clip	Permette agli utenti di visualizzare o di cercare le clip salvate nel canale di uno streamer	Interattiva	200 mila al giorno
18. Modifica del profilo	Modifica del profilo	Interattiva	5 mila al giorno
19. Modifica del canale	Permette agli streamer di modificare le informazioni del proprio canale	Interattiva	10 mila al giorno

20. Ottenere num medio spect. a live o il num di spettatori nella live corrente	Permette agli streamer di ottenere num medio spect. a live o il num di spettatori nella live corrente	Batch	1 volta al giorno
---	---	-------	--------------------------

2.3 Ristrutturazione dello schema ER

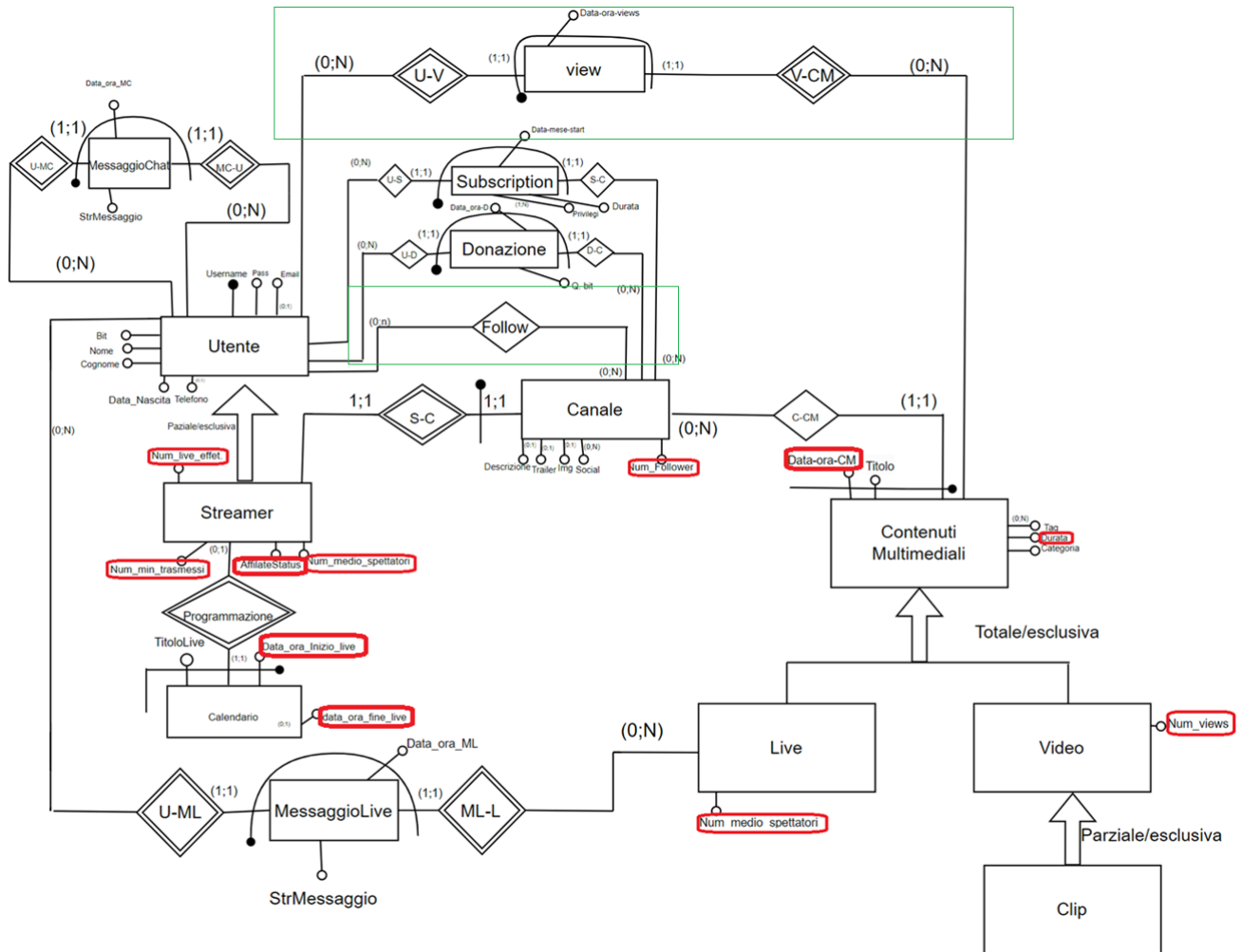
2.3.1 Analisi delle ridondanze

Ripensando alla memorizzazione di alcune statistiche degli streamer, possiamo adottare un approccio più efficiente. Piuttosto che memorizzare separatamente il numero di live effettuate, il numero di minuti trasmessi e il numero medio di spettatori simultanei per ogni streamer, possiamo utilizzare informazioni già disponibili per dedurre o calcolare queste statistiche:

➤ **Alcune ridondanze individuate sono le seguenti:**

- *Memorizzazione del numero di minuti trasmessi:*
 - **Convienne NON mantenere la ridondanza:**
Considerando che il numero di minuti trasmessi da uno streamer potrebbe essere considerato un dato relativamente irrilevante e che l'accesso frequente a questa informazione potrebbe non essere un requisito essenziale, potrebbe essere ragionevole calcolarlo dinamicamente utilizzando altre informazioni disponibili, come la durata delle live o le informazioni presenti nel calendario dello streamer. Può essere ottenuto moltiplicando la durata delle live (che può essere ricavato dalla generalizzazione di Contenuti Multimediali, quindi ogni live avrà un attributo durata, oppure lo posso dedurre dal calendario dello Streamer, andando a sommare le varied urate delle live, sapendo la data e l'ora di inizio e fine) per il numero totale di live effettuate. Questo evita la necessità di memorizzare esplicitamente il numero di minuti trasmessi per ogni streamer.
- *Memorizzazione dellla durata delle live:*
 - **Convienne mantenere la ridondanza:**
Calcolare dinamicamente la durata delle live ogni volta che viene richiesta potrebbe richiedere un tempo di calcolo significativo, specialmente se ci sono molte live nel sistema. Mantenendo l'attributo della durata delle live, è possibile accedere rapidamente a questa informazione senza dover effettuare calcoli aggiuntivi, e mantiene coerenza dei dati nel tempo
- *Memorizzazione del numero di follower di un canale:*
 - **Convienza mantenere la ridondanza:**
Mantenere la memorizzazione separata del numero di follower di un canale può essere conveniente per evitare il calcolo dinamico del numero di follower ogni volta che viene richiesto. Inoltre, può semplificare le operazioni di classificazione e di accesso alle informazioni sugli streamer più seguiti.
- *Memorizzazione del numero medio di spettatori per live e streamer:*
 - **Convienza mantenere la ridondanza:**
Mantenere la memorizzazione separata del numero medio di spettatori per live e per streamer potrebbe essere utile per migliorare le prestazioni del sistema. Calcolare questi valori dinamicamente richiederebbe l'elaborazione di tutti i dati delle live o degli streamer ad ogni richiesta, aumentando il tempo di accesso. Mantenendo la ridondanza, è possibile accedere rapidamente alle informazioni senza dover calcolare i valori ogni volta.

■ Analisi grafica delle possibili ridondanze



Analisi della ridondanza: “Ridondanza nella memorizzazione del numero di minuti trasmessi come attributo su Streamer”.

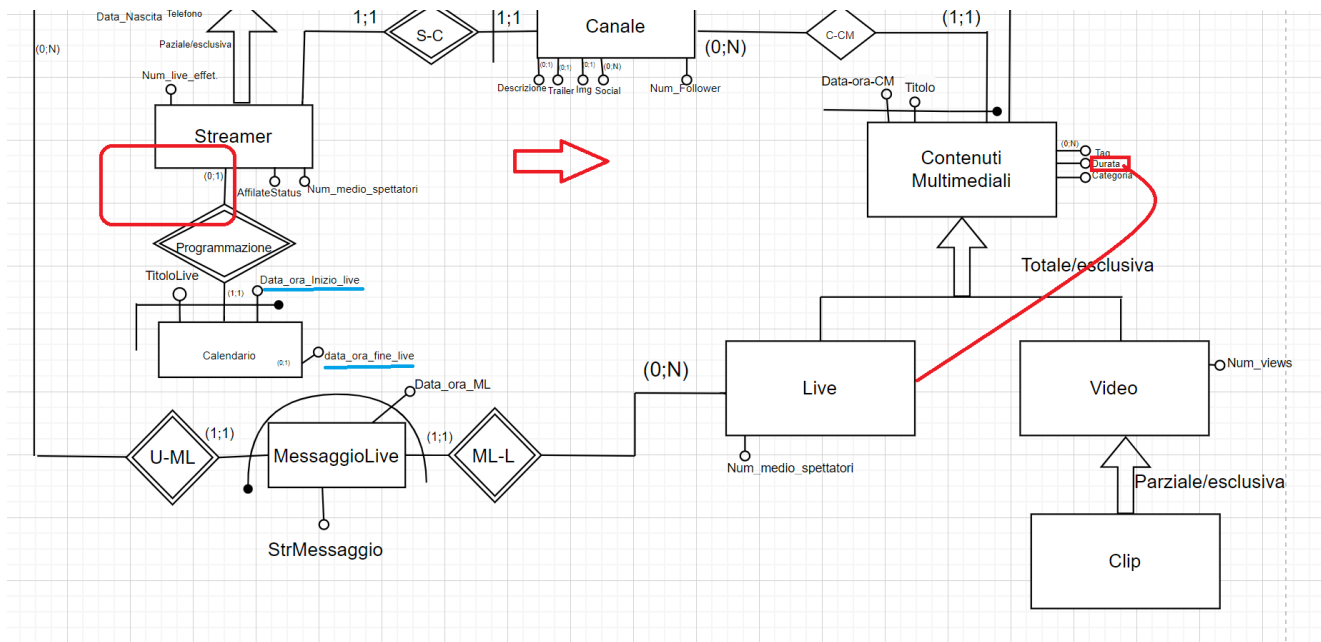
❖ Data l’operazione: Ottenere il numero di minuti trasmessi di uno streamer.

Tavola dei volume necessari:

Concetto	Tipo	Volume
Streamer	E	30 milioni
Contenuti Multimediali	E	200 milioni
Live	E	120 milioni
S-C	A	30 milioni
Canale	E	30 milioni
C-CM	A	200 milioni

■ Analisi senza ridondanza

➤ Schema di operazione



■ Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
Streamer	Entità	1	L
S-C	Associazione	1	L
canale	Entità	1	L
C-CM	Associazione	1	L
ContenutiMultimediali → Live	Entità	4	L

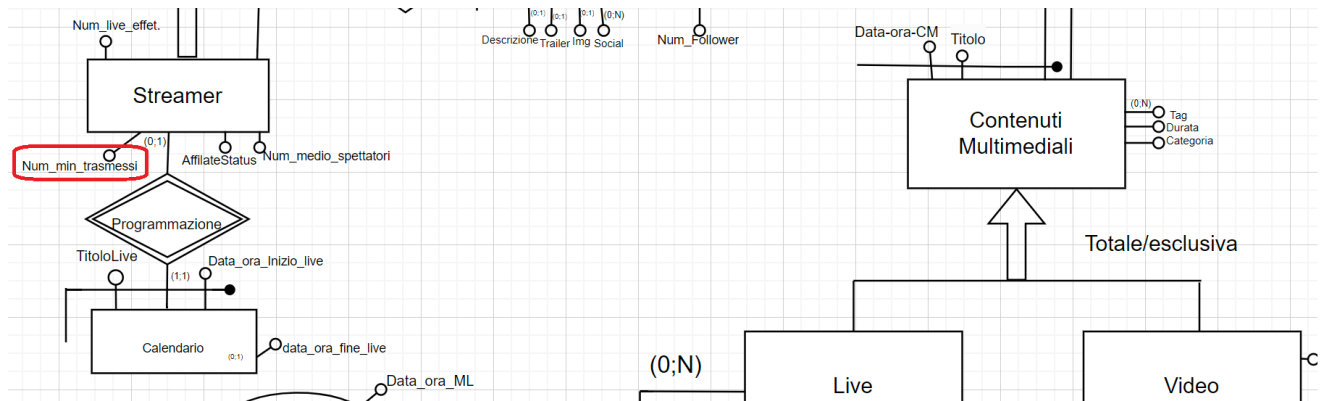
Media = Numero totale di contenuti multimediali di Live / Numero di streamer

Media = 120 milioni / 30 milioni = 4 .

Quindi, in media, ogni streamer ha circa 4 contenuti di Live effettuate.

■ Analisi con ridondanza

➤ Schema di operazione



■ Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
Streamer	Entità	1	L

➤ Analisi di complessità senza ridondanza

■ Spazio: 0 byte

■ Tempo: 8 accessi (al giorno)

- ❖ In assenza di ridondanza: L'occupazione dello spazio sarà inferiore poiché non è necessario salvare l'attributo "Num_min_trasmessi" in modo duplicato. Tuttavia, il tempo di accesso al numero di minuti trasmessi sarà maggiore in quanto è necessario calcolare dinamicamente il valore utilizzando le informazioni delle live.

➤ Analisi di complessità con ridondanza

■ Spazio: 4 byte * 30 milioni = 120 milioni di byte

■ Tempo: 1 accesso (al giorno)

- ❖ In presenza di ridondanza: L'occupazione dello spazio sarà maggiore a causa della presenza dell'attributo "Num_min_trasmessi" nell'entità Streamer, che richiede la memorizzazione dei dati in modo duplicato. Tuttavia, il tempo di accesso al numero di minuti trasmessi sarà ridotto poiché i dati sono disponibili direttamente come attributo.

❖ Scelta implementativa:

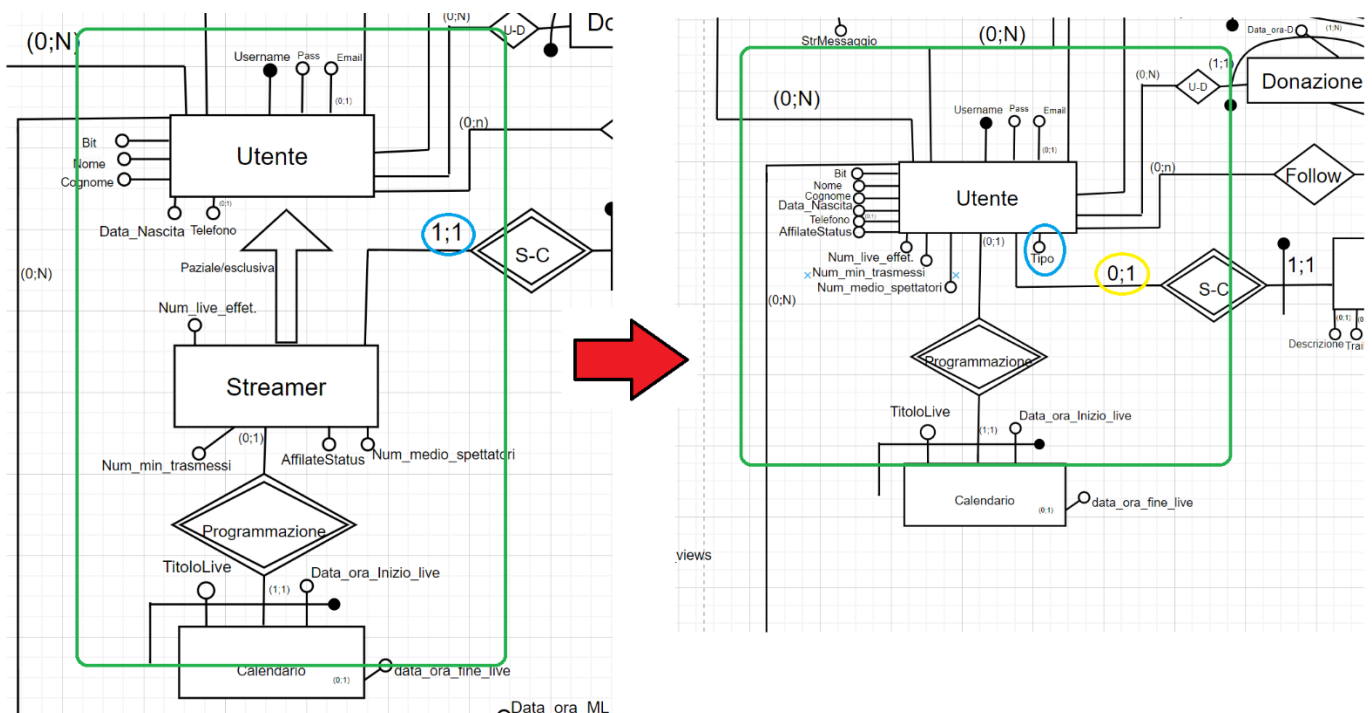
Dato che il numero di minuti trasmessi di uno streamer può variare nel tempo in base alle live trasmesse, è preferibile evitare la ridondanza e calcolare il valore in modo dinamico. In questo modo, il numero di minuti trasmessi sarà sempre aggiornato e rifletterà correttamente l'attività dello streamer. Anche se la ridondanza potrebbe ridurre il tempo di accesso ai dati, la priorità principale è mantenere l'accuratezza e l'aggiornamento delle informazioni. Calcolare il numero di minuti trasmessi dinamicamente utilizzando le informazioni delle live garantisce un'informazione più precisa e attuale, anche se richiede un po' più di tempo durante l'accesso, ma permette così di avere un database molto più "leggero" in termini di spazio occupato.

2.3.2 Eliminazione delle generalizzazioni

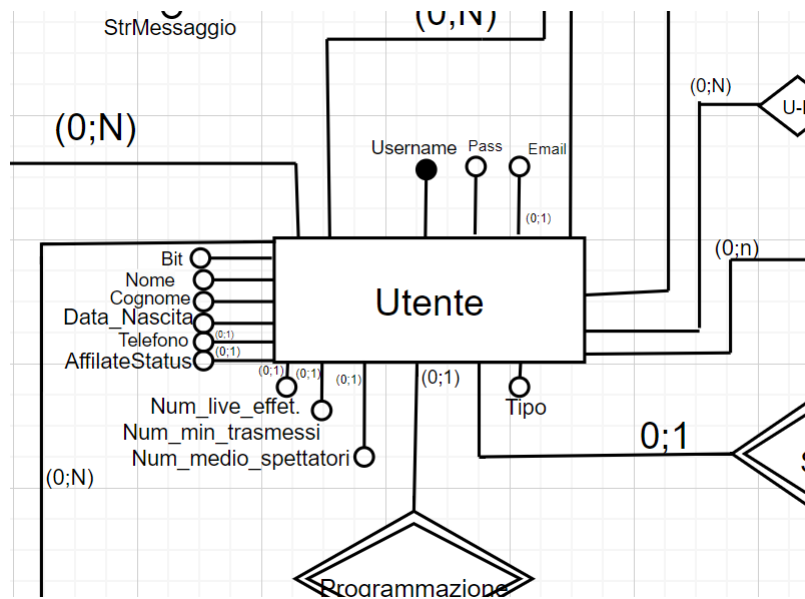
Sullo schema ER sono state ritrovate **tre generalizzazioni**:

- Di seguito verrà illustrato come si è deciso di rimuoverle:

■ Generalizzazione Utente → Streamer



- Si è deciso di raggruppare le entità figlie all'interno dell'entità genitore, aggiungendo quindi un attributo per distinguerle:



❖ Scelta implementativa:

Questa è una forma di generalizzazione nel modello ER che consente di gestire in modo più efficiente le operazioni comuni tra i diversi tipi di utenti.

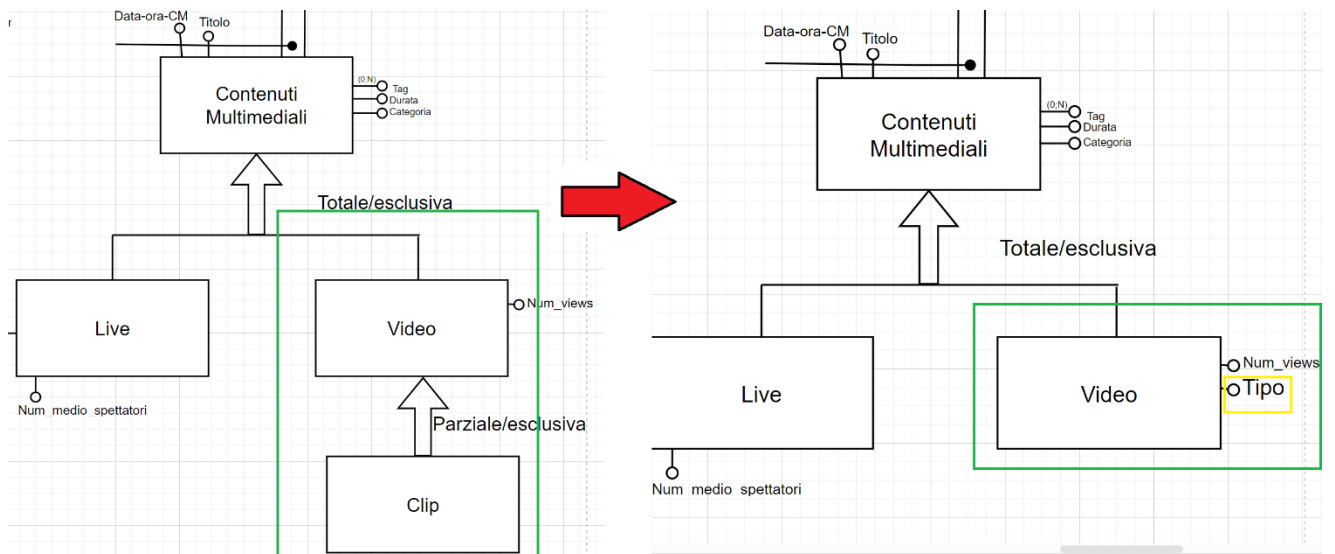
Gli Utenti Spettatori e gli Utenti Streamer hanno molti attributi in comune, come nome utente, password, email, e così via, può essere conveniente considerarli come un'unica entità "Utente".

L'attributo "tipo" viene utilizzato per distinguere tra gli Utenti Spettatori e gli Utenti Streamer (o entrambi) e consentire funzionalità specifiche in base al ruolo dell'utente.

Ciò semplifica la gestione dei dati e riduce la complessità del modello, in quanto le operazioni comuni possono essere eseguite sull'entità "Utente" senza la necessità di considerare separatamente gli Utenti Spettatori e gli Utenti Streamer.

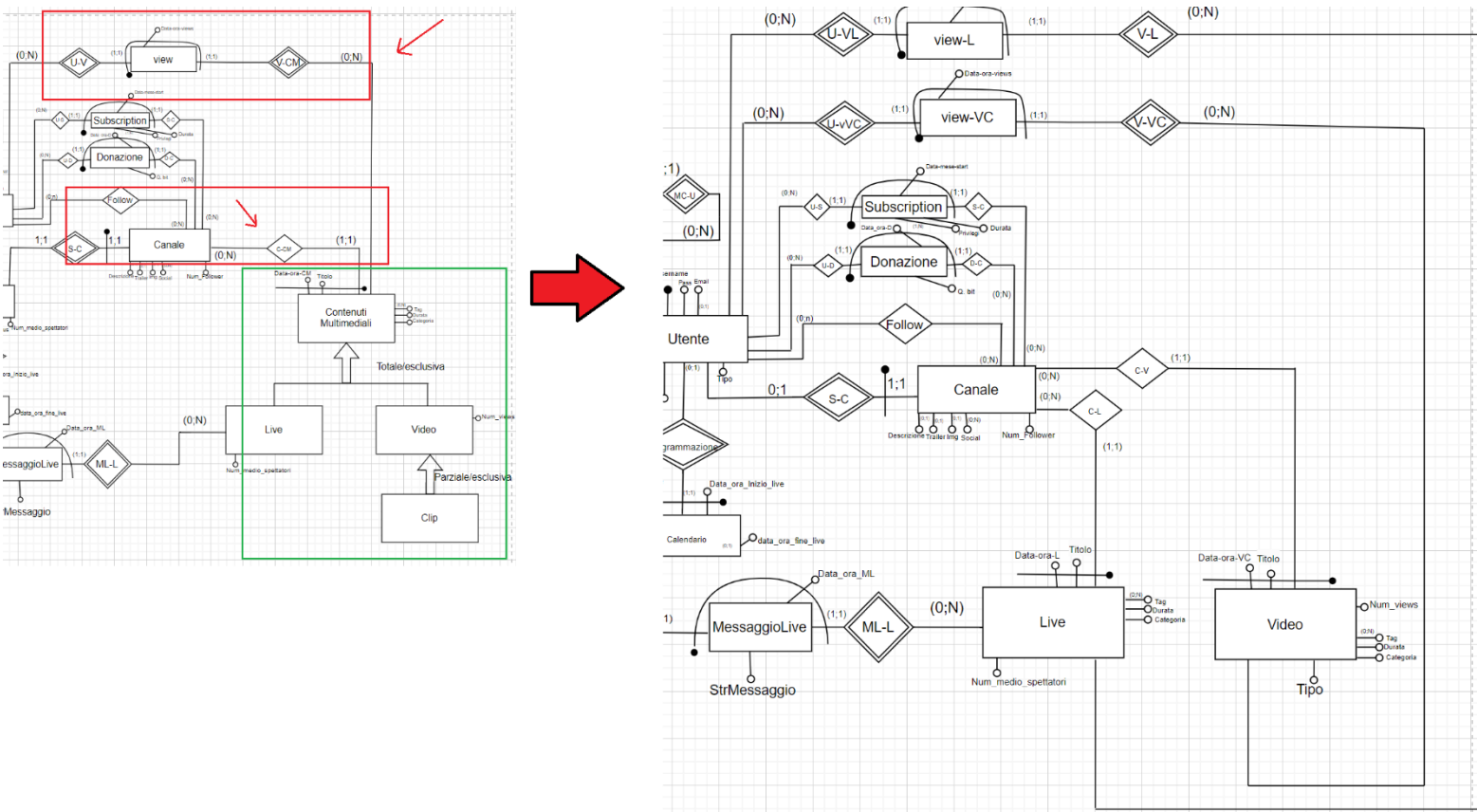
■ Generalizzazione video → Clip

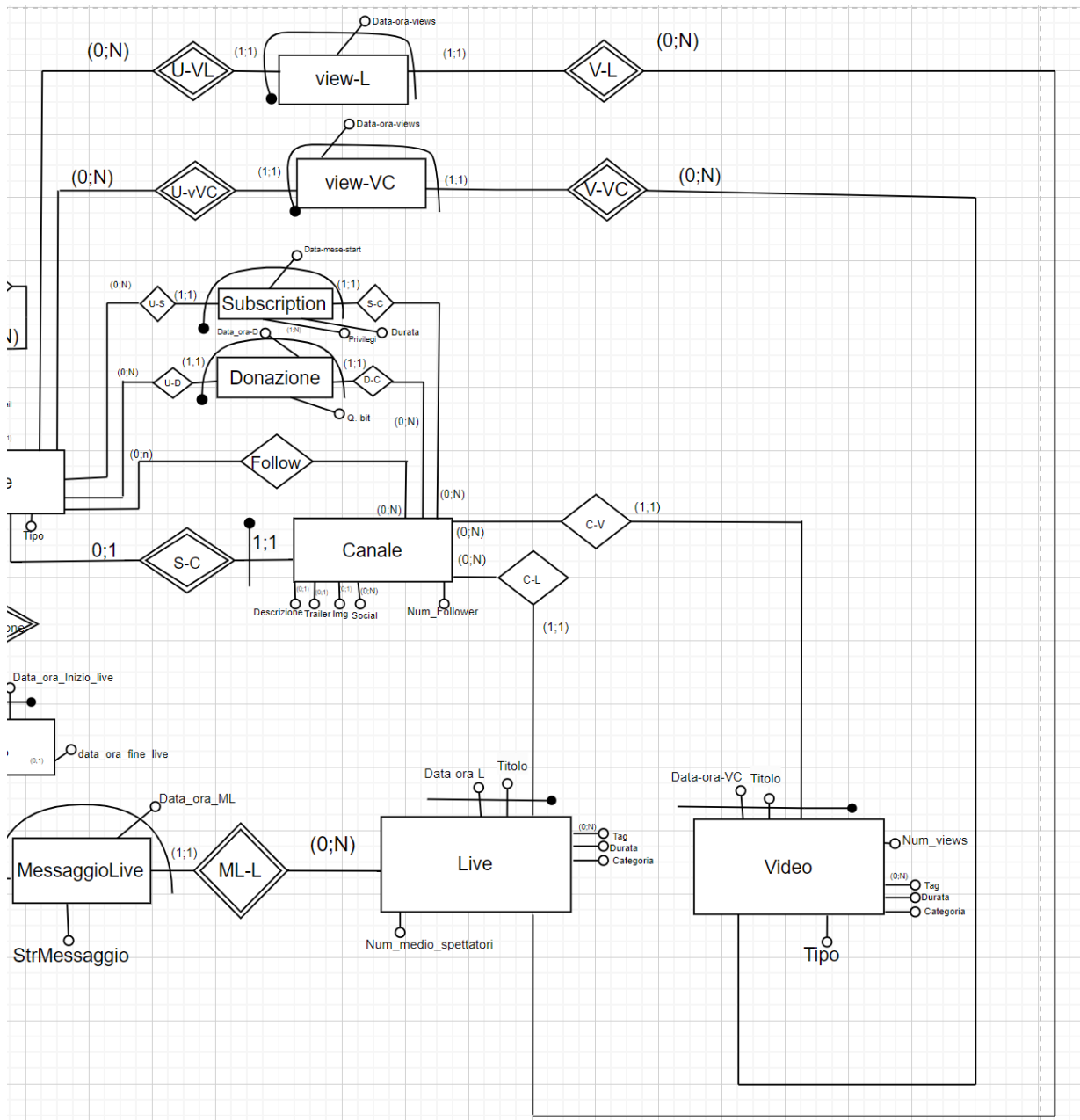
- Si è deciso di raggruppare le entità figlie all'interno dell'entità genitore, aggiungendo quindi un attributo per distinguerle:



■ Generalizzazione Contenuti Multimediali → Live ,Video →Clip

- Si è deciso di raggruppare l'entità del genitore all'interno dell'entità figlie.





❖ Scelta implementativa:

Le entità "Live" e "Video" rappresentano concetti "distinti". Una "Live" rappresenta una trasmissione in tempo reale, mentre un "Video" indica una registrazione di una live passata. Tenere separate queste entità permette una rappresentazione più accurata e semantica dei concetti specifici. Separare le entità "Live" e "Video" semplifica la gestione dei dati associati. Si può mantenere informazioni specifiche sulla live, come lo stato attuale, i dati in tempo reale e le interazioni degli utenti, nell'entità "Live". D'altra parte, l'entità "Video" può contenere informazioni relative alla registrazione della live passata come le statistiche di visualizzazione. Questo rende la gestione dei dati più chiara e coerente.

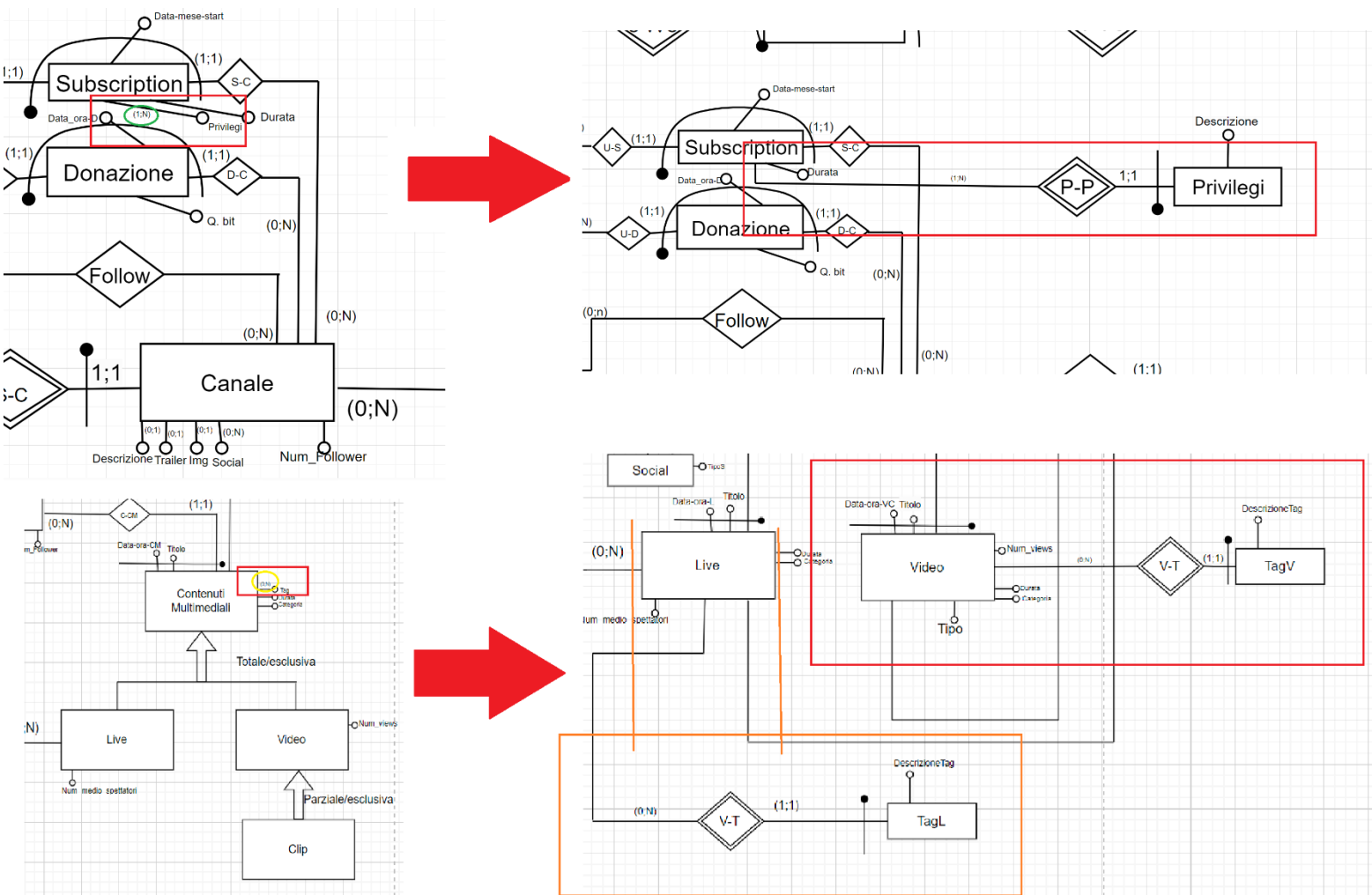
In sintesi, la separazione delle entità "Live" e "Video", consente di modellare in modo più preciso i concetti distinti, e di gestire i dati in modo più strutturato e di rappresentare in modo chiaro le relazioni tra le entità.

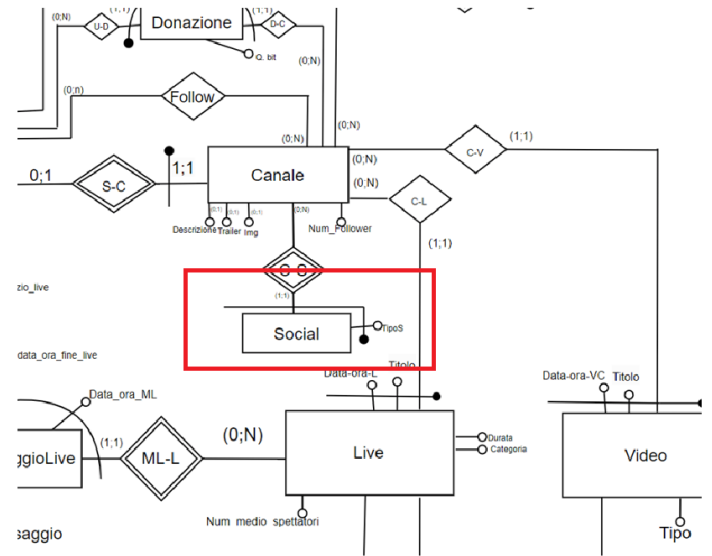
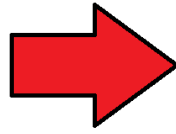
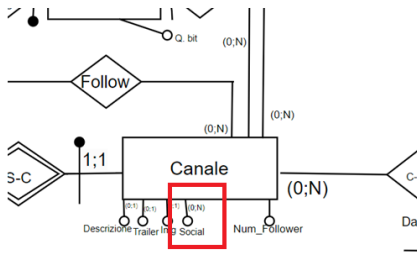
2.3.3 Eventuale eliminazione degli attributi composti e degli attributi multivalore

➤ Normalizzazione degli attributi multipli

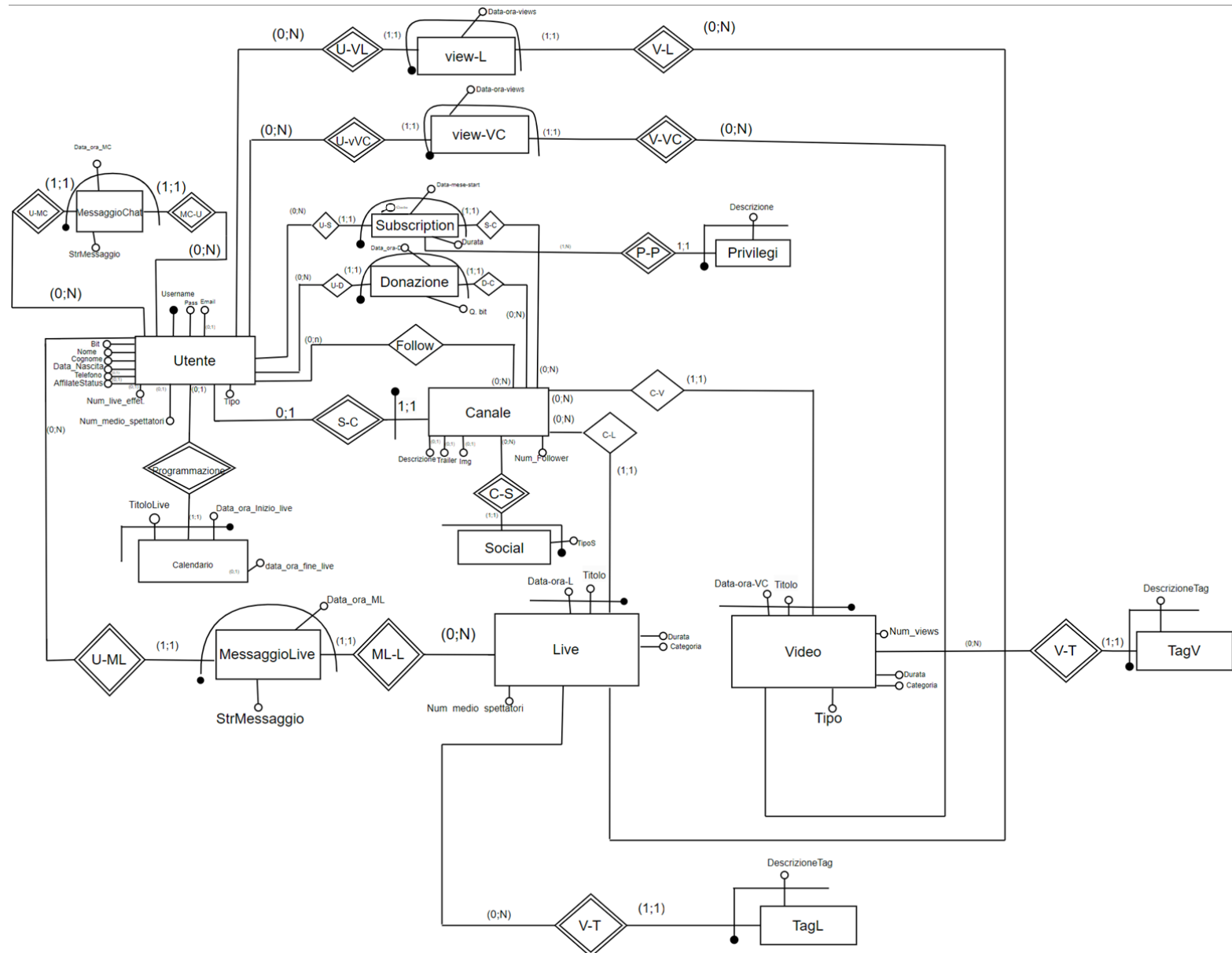
1. **Attributo "Privilegi"** ➔ I viewer possono supportare gli streamer tramite la subscription (a pagamento) al loro canale, ottenendo dei privilegi (emoticon personalizzate, ecc.)
2. **Attributo "Tag"** ➔ diversi tag per ogni Live, video o clip
3. **Attributo Social**

✓ Trasformo gli attributi multivalore, reificando l'attributo e aggiungendo un'associazione:





2.4 Schema E-R ristrutturato + business rules



Business rules E-R ristrutturato:

1. Tipo di Utenti → SS=StreamerSpettatore,Spettatore,Streamer
2. Solo utente di Tipo Streamer possiede canale , e di conseguenza può effettuare Live.
3. L'utente di Tipo Spettatore può guardare Live/video/clip,interagire con la chat Video e mandare messaggi ad altri utenti.
4. Solo utente di Tipo Streamer possiede le statistiche da Streamer(NumLiveEff.,AffiliateStatus,NumMedioSpect.,...)
5. Lo Streamer inizia la Live, che viene trasmessa nel canale dello streamer.
6. La live, può diventare Video sul canale, le clip sono video brevi di live passate.
7. Un utente che sia,streamer e anche spettatore, non può autodonarsi o autoabbonarsi al suo stesso canale con lo stesso account.
8. Un utente non può mandarsi messaggi da solo.
9. Sia live che video(di tipo Video e Clip) possono avere dei tag
- 10.Un utente possiede calendario solo se è di tipo Streamer o SS.

2.5 Schema relazionale

_ → PK.

Utente(Username, Nome, Cognome, password, email*, DataNascita, Telefono*, Tipo, AffiliateStatus*, NumliveEff.*, NumMedioSpettatori*, Bit)

Canale(Descrizione*, Trailer*, Img*, NumFollower, UsernameUtente)

Social(TipoS, UsernameCanaleUtente)

Live(Titolo, DataOraL, UsernameCanaleUtente, Durata, Categoria, NumMedioSpettatori) +

Video(DataOraVc, Titolo, UsernameCanaleUtente, Durata, categoria, NumMedioSpettatori, Tipo)

TagV(DescrizioneTag, DataOraVc, UsernameCanaleUtente, TitoloVideo)

TagL(DescrizioneTag, DataOraL, UsernameCanaleUtente, TitoloLive) +

Calendario(TitoloLive, DataOraInizioLive, UsernameUtente, DataOraFineLive*) +

MessaggioChat(UsernameUtenteMittente, UsernameUtenteRicevente, DataOraMcInvio, StrMessaggio)

MessaggioLive(UsernameUtenteMittente, DataOraMLInvio, UsernameCanaleUtente, DataOraL, TitoloLive, StrMessaggio) +

Donazioni(UsernameUtente, UsernameUtenteCanale, DataOraDonazione, Q.bit)

Subscription(UsernameUtente, UsernameUtenteCanale, DataMeseInizio, Durata, DescrizionePrivilegi, Costo)

Privilegi(Descrizione, UsernameUtente, UsernameUtenteCanale, DataMeseInizio)

ViewL(UsernameUtente, UsernameUtenteCanale, DataOraL, DataOraViews, TitoloLive) +

ViewVC(UsernameUtente, UsernameUtenteCanale, DataOraV-C, TitoloV-C, DataOraViews)

Follow(UsernameUtente, UsernameCanaleUtente)

N.B. + ← Modifica chiave primaria , si è voluto , dopo un analisi approfondita, apportare una semplificazione della chiave primaria, per renderla più minimale possibile e togliere anomalie.

Indicazione dei vincoli di integrità referenziale:

- *Canale (UsernameUtente)referenzia Utente(Username)*
- *Social (UsernameCanaleUtente) referenzia Canale(UsernameUtente)*
- *Live (UsernameCanaleUtente) referenzia Canale(UsernameUtente)*
- *Video (UsernameCanaleUtente) referenzia Canale(UsernameUtente)*
- *TagV (UsernameCanaleUtente) referenzia Video(UsernameCanaleUtente)*
- *TagV (DataOraVC) referenzia Video(DataOraVC)*
- *TagV (TitoloVideo)referenzia Video(Titolo)*
- *TagL (UsernameCanaleUtente) referenzia Live(UsernameCanaleUtente)*
- *TagL (DataOraL) referenzia Live(DataOraL)*
- *TagL (TitoloLive)referenzia Live(Titolo)*
- *Calendario (UsernameUtente) referenzia Utente(Username)*
- *MessaggioChat (UsernameUtenteMittente) referenzia Utente(Username)*
- *MessaggioChat (UsernameUtenteRicevente) referenzia Utente(Username)*
- *MessaggioLive (UsernameUtenteMittente) referenzia Utente(Username)*
- *MessaggioLive (UsernameCanaleUtente) referenzia Canale(Utente)*
- *MessaggioLive (DataOraL) referenzia Live(DataOraL)*
- *MessaggioLive (TitoloLive) referenzia Live(Titolo)*
- *Donazione (UsernameUtente) referenzia Utente(Username)*
- *Donazione (UsernameCanaleUtente) referenzia Canale(Username)*
- *Subscription (UsernameUtente) referenzia Utente(Username)*
- *Subscription (UsernameCanaleUtente) referenzia Canale(Username)*
- *Privilegi (UsernameUtente) referenzia Subscription (UsernameUtente)*
- *Privilegi (UsernameCanaleUtente) referenzia UsernameCanaleUtente (UsernameUtente)*
- *Follow (UsernameUtente) referenzia Utente(Username)*
- *Follow (UsernameCanaleUtente) referenzia Canale(Username)*
- *ViewL (UsernameCanaleUtente) referenzia UsernameCanaleUtente (Live)*
- *ViewL (UsernameUtente) referenzia Utente(Username)*
- *ViewL (TitoloLive) referenzia Live(Titolo)*
- *ViewL (DataOraLive) referenzia Live(DataOraL)*
- *ViewVC (UsernameCanaleUtente) referenzia Video(UsernameCanaleUtente)*
- *ViewL (UsernameUtente) referenzia Utente(Username)*
- *ViewL (TitoloV-C) referenzia Video(Titolo)*
- *ViewL (DataOraV-C) referenzia Video(DataOraV-C)*

3.0 Implementazione SQL

3.1 DDL di creazione del database.

```

1. create database TwitchDBProject

1. CREATE TABLE Utente (
2.   Username VARCHAR(255) PRIMARY KEY,
3.   Nome VARCHAR(255) NOT NULL,
4.   Cognome VARCHAR(255) NOT NULL,
5.   Password VARCHAR(255) NOT NULL,
6.   Email VARCHAR(255),
7.   DataNascita DATE,
8.   Telefono VARCHAR(255),
9.   Tipo VARCHAR(20) CHECK (Tipo IN ('UtenteStreamer', 'UtenteSpettatore', 'UtenteSS')),
10.  AffiliateStatus BIT,
11.  NumliveEff INT,
12.  NumMedioSpettatori INT,
13.  QBIT INT,
14.
15.  CONSTRAINT Utente_Check CHECK (
16.    (Tipo = 'UtenteSpettatore' AND AffiliateStatus IS NULL AND NumliveEff IS NULL AND
NumMedioSpettatori IS NULL) OR
17.    (Tipo <> 'UtenteSpettatore' AND AffiliateStatus IS NOT NULL AND NumliveEff IS NOT NULL AND
NumMedioSpettatori IS NOT NULL)
18.  )
19. );
20.
21. CREATE TABLE Canale (
22.  Descrizione VARCHAR(255),
23.  Trailer VARCHAR(255),
24.  Img VARCHAR(255),
25.  NumFollower INT,
26.  UsernameUtente VARCHAR(255),
27.  PRIMARY KEY (UsernameUtente),
28.  FOREIGN KEY (UsernameUtente) REFERENCES Utente (Username)
29. );
30.
31. CREATE TABLE Social (
32.  TipoS VARCHAR(255),
33.  UsernameUtente VARCHAR(255),
34.  PRIMARY KEY (TipoS, UsernameUtente),
35.  FOREIGN KEY (UsernameUtente) REFERENCES Utente (Username)
36. );
37.
38. CREATE TABLE Live (
39.  Titolo VARCHAR(255),
40.  DataOral TIMESTAMP,
41.  UsernameUtente VARCHAR(255),
42.  Durata INT,
43.  Categoria VARCHAR(255),
44.  NumMedioSpettatori INT,
45.  PRIMARY KEY (Titolo, DataOral, UsernameUtente),
46.  FOREIGN KEY (UsernameUtente) REFERENCES Canale (UsernameUtente)
47. );
48.
49. CREATE TABLE Video (
50.  DataOraVc TIMESTAMP,
51.  Titolo VARCHAR(255),
52.  UsernameUtente VARCHAR(255),
53.  Durata INT,
54.  Categoria VARCHAR(255),
55.  NumMedioSpettatori INT,
56.  Tipo VARCHAR(20) CHECK (Tipo IN ('Video', 'Clip')),
57.  PRIMARY KEY (DataOraVc, Titolo, UsernameUtente),
58.  FOREIGN KEY (UsernameUtente) REFERENCES Canale (UsernameUtente)
59. );
60.

```

```

61. CREATE TABLE TagV (
62.     DescrizioneTag VARCHAR(255),
63.     DataOraVc TIMESTAMP,
64.     TitoloVideo VARCHAR(255),
65.     UsernameUtente VARCHAR(255),
66.     PRIMARY KEY (DescrizioneTag, DataOraVc, TitoloVideo, UsernameUtente),
67.     FOREIGN KEY (DataOraVc, TitoloVideo, UsernameUtente) REFERENCES Video (DataOraVc, Titolo,
UsernameUtente)
68. );
69.
70. CREATE TABLE TagL (
71.     DescrizioneTag VARCHAR(255),
72.     DataOraL TIMESTAMP,
73.     TitoloLive VARCHAR(255),
74.     UsernameUtente VARCHAR(255),
75.     PRIMARY KEY (DescrizioneTag, DataOraL, TitoloLive, UsernameUtente),
76.     FOREIGN KEY (DataOraL, TitoloLive, UsernameUtente) REFERENCES Live (DataOraL, Titolo,
UsernameUtente)
77. );
78.
79. CREATE TABLE Calendario (
80.     TitoloLive VARCHAR(255),
81.     DataOraInizioLive TIMESTAMP,
82.     UsernameUtente VARCHAR(255),
83.     DataOraFineLive TIMESTAMP,
84.     PRIMARY KEY (TitoloLive, DataOraInizioLive, UsernameUtente),
85.     FOREIGN KEY (UsernameUtente) REFERENCES Utente (Username)
86. );
87.
88. CREATE TABLE MessaggioChat (
89.     UsernameUtenteMittente VARCHAR(255),
90.     UsernameUtenteRicevente VARCHAR(255),
91.     DataOraMcInvio TIMESTAMP,
92.     StrMessaggio TEXT,
93.     PRIMARY KEY (UsernameUtenteMittente, UsernameUtenteRicevente, DataOraMcInvio),
94.     FOREIGN KEY (UsernameUtenteMittente) REFERENCES Utente (Username),
95.     FOREIGN KEY (UsernameUtenteRicevente) REFERENCES Utente (Username)
96. );
97.
98. CREATE TABLE MessaggioLive (
99.     UsernameUtenteMittente VARCHAR(255),
100.    DataOraMLInvio TIMESTAMP,
101.    UsernameCanaleUtente VARCHAR(255),
102.    DataOraL TIMESTAMP,
103.    TitoloLive VARCHAR(255),
104.    StrMessaggio TEXT,
105.    PRIMARY KEY (UsernameUtenteMittente, DataOraMLInvio, UsernameCanaleUtente, DataOraL, TitoloLive),
106.    FOREIGN KEY (UsernameUtenteMittente) REFERENCES Utente (Username),
107.    FOREIGN KEY (UsernameCanaleUtente) REFERENCES Canale (UsernameUtente),
108.    FOREIGN KEY (DataOraL, TitoloLive, UsernameCanaleUtente) REFERENCES Live (DataOraL, Titolo,
UsernameUtente)
109. );
110.
111.
112. CREATE TABLE Donazioni (
113.     UsernameUtente VARCHAR(255),
114.     UsernameUtenteCanale VARCHAR(255),
115.     DataOraDonazione TIMESTAMP,
116.     QBIT INT,
117.     PRIMARY KEY (UsernameUtente, UsernameUtenteCanale, DataOraDonazione),
118.     FOREIGN KEY (UsernameUtente) REFERENCES Utente (Username),
119.     FOREIGN KEY (UsernameUtenteCanale) REFERENCES Canale (UsernameUtente)
120. );
121.
122. CREATE TABLE Subscription (
123.     UsernameUtente VARCHAR(255),
124.     UsernameUtenteCanale VARCHAR(255),
125.     DataMeseInizio TIMESTAMP,
126.     Durata INT,
127.     DescrizionePrivilegi VARCHAR(255),
128.     PRIMARY KEY (UsernameUtente, UsernameUtenteCanale, DataMeseInizio),

```

```

129. FOREIGN KEY (UsernameUtente) REFERENCES Utente (Username),
130. FOREIGN KEY (UsernameUtenteCanale) REFERENCES Canale (UsernameUtente)
131. );
132.
133. CREATE TABLE Privilegi (
134.     Descrizione VARCHAR(255),
135.     UsernameUtente VARCHAR(255),
136.     UsernameUtenteCanale VARCHAR(255),
137.     DataMeseInizio TIMESTAMP,
138.     PRIMARY KEY (Descrizione, UsernameUtente, UsernameUtenteCanale, DataMeseInizio),
139.     FOREIGN KEY (UsernameUtente, UsernameUtenteCanale, DataMeseInizio) REFERENCES Subscription
140.     (UsernameUtente, UsernameUtenteCanale, DataMeseInizio)
141. );
142.
143. CREATE TABLE ViewL (
144.     UsernameUtente VARCHAR(255),
145.     UsernameUtenteCanale VARCHAR(255),
146.     DataOral TIMESTAMP,
147.     TitoloLive VARCHAR(255),
148.     DataOraViews TIMESTAMP,
149.     PRIMARY KEY (UsernameUtente, UsernameUtenteCanale, DataOral, TitoloLive, DataOraViews),
150.     FOREIGN KEY (UsernameUtente) REFERENCES Utente (Username),
151.     FOREIGN KEY (UsernameUtenteCanale) REFERENCES Canale (UsernameUtente),
152.     FOREIGN KEY (DataOral, TitoloLive, UsernameUtente) REFERENCES Live (DataOral, Titolo,
153.     UsernameUtente),
154.     UNIQUE (UsernameUtente, UsernameUtenteCanale, DataOral, TitoloLive, DataOraViews)
155. );
156.
157.
158.
159. CREATE TABLE ViewVC (
160.     UsernameUtente VARCHAR(255),
161.     UsernameCanaleUtente VARCHAR(255),
162.     DataOraVC TIMESTAMP,
163.     TitoloVC VARCHAR(255),
164.     DataOraViews TIMESTAMP,
165.     PRIMARY KEY (UsernameUtente, UsernameCanaleUtente, DataOraVC, TitoloVC, DataOraViews),
166.     FOREIGN KEY (UsernameUtente) REFERENCES Utente (Username),
167.     FOREIGN KEY (UsernameCanaleUtente) REFERENCES Canale (UsernameUtente),
168.     FOREIGN KEY (DataOraVC, TitoloVC, UsernameCanaleUtente) REFERENCES Video (DataOraVC, Titolo,
169.     UsernameUtente),
170.     UNIQUE (UsernameUtente, UsernameCanaleUtente, DataOraVC, TitoloVC, DataOraViews)
171. );
172.
173.
174. CREATE TABLE Follow (
175.     UsernameUtente VARCHAR(255),
176.     UsernameCanaleUtente VARCHAR(255),
177.     PRIMARY KEY (UsernameUtente, UsernameCanaleUtente),
178.     FOREIGN KEY (UsernameUtente) REFERENCES Utente (Username),
179.     FOREIGN KEY (UsernameCanaleUtente) REFERENCES Canale (UsernameUtente)
180. );
181.

```


3.2 DML di popolamento di tutte le tabelle del database.

```

1. -- Inserimenti nella tabella Utente
2. INSERT INTO Utente (Username, Nome, Cognome, Password, Email, DataNascita, Telefono, Tipo,
affiliatestatus, NumliveEff, NumMedioSpettatori, qbit)
3. VALUES ('user1', 'Mario', 'Rossi', 'password1', 'mario@email.com', '1990-01-01', '123456789',
'UtenteSpettatore', NULL, NULL, NULL, 0);
4.
5. INSERT INTO Utente (Username, Nome, Cognome, Password, Email, DataNascita, Telefono, Tipo,
affiliatestatus, NumliveEff, NumMedioSpettatori, qbit)
6. VALUES ('user2', 'Laura', 'Verdi', 'password2', 'laura@email.com', '1995-05-10', '987654321',
'UtenteStreamer', CAST(1 AS bit), 10, 500, 0);
7.
8. -- Inserimenti nella tabella Canale
9. INSERT INTO Canale (Descrizione, Trailer, Img, NumFollower, UsernameUtente)
10. VALUES ('Canale di Mario', 'trailer_url1', 'img_url1', 1000, 'user1');
11.
12. INSERT INTO Canale (Descrizione, Trailer, Img, NumFollower, UsernameUtente)
13. VALUES ('Canale di Laura', 'trailer_url2', 'img_url2', 5000, 'user2');
14.
15. -- Inserimenti nella tabella Social
16. INSERT INTO Social (TipoS, UsernameUtente)
17. VALUES ('Facebook', 'user1');
18.
19. INSERT INTO Social (TipoS, UsernameUtente)
20. VALUES ('Instagram', 'user2');
21.
22. -- Inserimenti nella tabella Live
23. INSERT INTO Live (Titolo, DataOraL, UsernameUtente, Durata, Categoria, NumMedioSpettatori)
24. VALUES ('Live 1', '2023-05-24 10:00:00', 'user1', 120, 'Gaming', 1000);
25.
26. INSERT INTO Live (Titolo, DataOraL, UsernameUtente, Durata, Categoria, NumMedioSpettatori)
27. VALUES ('Live 2', '2023-05-25 15:30:00', 'user2', 90, 'Music', 500);
28.
29. -- Inserimenti nella tabella Video
30. INSERT INTO Video (DataOraVC, Titolo, UsernameUtente, Durata, Categoria, NumMedioSpettatori, Tipo)
31. VALUES ('2023-05-26 12:00:00', 'Video 1', 'user1', 180, 'VideoLP', 800, 'Video');
32.
33. INSERT INTO Video (DataOraVC, Titolo, UsernameUtente, Durata, Categoria, NumMedioSpettatori, Tipo)
34. VALUES ('2023-05-27 18:45:00', 'Video 2', 'user2', 240, 'clip', 2000, 'Clip');
35.
36. -- Inserimenti nella tabella TagV
37. INSERT INTO TagV (DescrizioneTag, DataOraVC, TitoloVideo, UsernameUtente)
38. VALUES ('Tag1', '2023-05-26 12:00:00', 'Video 1', 'user1');
39.
40. INSERT INTO TagV (DescrizioneTag, DataOraVC, TitoloVideo, UsernameUtente)
41. VALUES ('Tag2', '2023-05-27 18:45:00', 'Video 2', 'user2');
42.
43. -- Inserimenti nella tabella TagL
44. INSERT INTO TagL (DescrizioneTag, DataOraL, TitoloLive, UsernameUtente)
45. VALUES ('Tag3', '2023-05-24 10:00:00', 'Live 1', 'user1');
46.
47. INSERT INTO TagL (DescrizioneTag, DataOraL, TitoloLive, UsernameUtente)
48. VALUES ('Tag4', '2023-05-25 15:30:00', 'Live 2', 'user2');
49.
50. -- Inserimenti nella tabella Calendario
51. INSERT INTO Calendario (TitoloLive, DataOraInizioLive, UsernameUtente, DataOraFineLive)
52. VALUES ('Live 1', '2023-05-24 10:00:00', 'user1', '2023-05-24 12:00:00');
53.
54. INSERT INTO Calendario (TitoloLive, DataOraInizioLive, UsernameUtente, DataOraFineLive)
55. VALUES ('Live 2', '2023-05-25 15:30:00', 'user2', '2023-05-25 17:00:00');
56.

```

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

```
Delete from Utente where username='user1'|
```

Data Output	Messages	Notifications
ERROR: ERRORE: l'istruzione UPDATE o DELETE sulla tabella "utente" viola il vincolo di chiave esterna "canale_usernameutente_fkey" sulla tabella "canale"		
DETAIL: La chiave (username)=(user1) è ancora referenziata dalla tabella "canale".		

- L'errore riscontrato evidenzia un tentativo di effettuare un'operazione di aggiornamento o eliminazione su una riga nella tabella "Utente" che presenta una chiave primaria (username) ancora referenziata nella tabella "Canale".
Ciò implica che nella tabella "Canale" esiste almeno una riga che contiene un valore nella colonna "UsernameUtente" corrispondente a "user1".
Al fine di poter procedere con successo all'operazione di aggiornamento o eliminazione, è necessario prima rimuovere o modificare le righe corrispondenti nella tabella "Canale" in modo che non facciano più riferimento alla chiave primaria (username) presente nella tabella "Utente".

```
INSERT INTO Utente (Username, Nome, Cognome, Password, Email, DataNascita, Telefono, Tipo, affilatestatus, NumliveEff, NumMedioSpettatori, qbit)
VALUES ('user3', 'Mario', 'Verdi', 'password3', 'mario@email.com', '1990-01-01', '123456789', 'UtenteSS', NULL, NULL, NULL, 0);
```

Data Output Messages Notifications

ERROR: ERRORE: la nuova riga per la relazione "utente" viola il vincolo di controllo "utente_check"

DETAIL: La riga in errore contiene (user3, Mario, Verdi, password3, mario@email.com, 1990-01-01, 123456789, UtenteSS, null, null, null, 0).

- L'errore riscontrato indica che la nuova riga che si sta cercando di inserire nella tabella "Utente" viola il vincolo di controllo denominato "utente_check".

Nello specifico, la riga errata contiene i seguenti valori: (user3, Mario, Verdi, password3, mario@email.com, 1990-01-01, 123456789, UtenteSS, null, null, null, 0).

Il vincolo di controllo "utente_check" specifica delle condizioni che devono essere rispettate per i valori della colonna "Tipo" nella tabella "Utente". In particolare, il vincolo richiede che:

- Se il valore della colonna "Tipo" è "UtenteSpettatore", allora le colonne "AffiliateStatus", "NumliveEff.", "NumMedioSpettatori" devono essere null.
- Altrimenti, se il valore della colonna "Tipo" non è "UtenteSpettatore", allora le colonne "AffiliateStatus", "NumliveEff.", "NumMedioSpettatori" non devono essere null.

Nella riga in questione, il valore della colonna "Tipo" è "UtenteSS" e le colonne "AffiliateStatus", "NumliveEff.", "NumMedioSpettatori" sono tutte null. Questa configurazione viola il vincolo "utente_check", poiché il valore della colonna "Tipo" non è "UtenteSpettatore", ma le colonne correlate sono null invece di essere valorizzate.

Per risolvere l'errore, è necessario correggere i valori nella riga da inserire in modo che rispettino le condizioni imposte dal vincolo di controllo "utente_check".

```
delete from Canale where usernameutente='user1'
```

Data Output Messages Notifications

ERROR: ERRORE: l'istruzione UPDATE o DELETE sulla tabella "canale" viola il vincolo di chiave esterna "live_usernameutente_fkey" sulla tabella "live"

DETAIL: La chiave (usernameutente)=(user1) è ancora referenziata dalla tabella "live".

- L'errore riscontrato indica che l'istruzione di aggiornamento o eliminazione che si è tentato di eseguire sulla tabella "Canale" viola il vincolo di chiave esterna denominato "live_usernameutente_fkey" sulla tabella "Live".

Nello specifico, la chiave (usernameutente) con il valore "user1" che si sta cercando di modificare o eliminare nella tabella "Canale" è ancora referenziata da almeno una riga nella tabella "Live".

Il vincolo di chiave esterna "live_usernameutente_fkey" garantisce l'integrità referenziale tra le tabelle "Canale" e "Live", richiedendo che ogni valore nella colonna "UsernameUtente" nella tabella "Live" corrisponda a un valore valido nella colonna "UsernameUtente" nella tabella "Canale".

Poiché la chiave (usernameutente) con valore "user1" è ancora referenziata nella tabella "Live", l'operazione di aggiornamento o eliminazione non può essere eseguita senza violare il vincolo di chiave esterna.