

# 1 PROGETTAZIONE CONCETTUALE

## 1.1. Requisiti iniziali

- **Requisiti consegna**  
Consegnare una relazione che illustri il lavoro effettuato per la progettazione della base di dati derivante dai requisiti indicati nella sezione “Requisiti iniziali” e che segua rigorosamente lo schema guida indicato nei file “2 Schema progettazione concettuale”, “3 Schema progettazione logica”, “4 DDL e DML”. Nella prima pagina della relazione devono essere riportati i nomi, cognomi, numeri di matricola e indirizzi e-mail dei componenti del gruppo. Supponendo che un gruppo sia composto dagli studenti di cognome A, B, C, si consegnino i seguenti file:
  - Un file “A\_B\_C\_Relazione” contenente la relazione (unione dei file “2 Schema progettazione concettuale”, “3 Schema progettazione logica”, “4 DDL e DML”, che non dovete consegnare separatamente).
  - Tre file “A\_B\_C\_DDL.sql”, “A\_B\_C\_DMLPOP.sql”, “A\_B\_C\_DMLUPD.sql” contenenti gli script SQL indicati nel file “4 DDL e DML”.
- Tali file dovranno essere consegnati in una cartella compressa “A\_B\_C.zip” insieme al file contenente le query della parte di SQL.

### Schema guida

1. Progettazione concettuale
  1. Requisiti iniziali (testo integrato con osservazioni fatte a partire dai file forniti)
  2. Glossario dei termini
  3. Requisiti riscritti
  4. Requisiti strutturati in gruppi di frasi omogenee
  5. Schema E-R + regole aziendali

Come auto-valutare il proprio schema E-R (+ business rules):

- **Correttezza:** controllare se i costrutti sono usati propriamente. Inoltre nella stesura dello schema E-R non bisogna considerare come verrà tradotto in relazionale (evitate errori come: omettere gli identificatori delle entità, aggiungere identificatori alle associazioni, aggiungere alle associazioni gli identificatori delle entità coinvolte, non indicare il tipo di generalizzazione, dare lo stesso nome a due entità o associazioni, usare un identificatore esterno basato su associazioni non (1,1) o un identificatore basato su attributi opzionali o multivalore)

- Completezza: rileggere i requisiti iniziali e considerare se ogni informazione rilevante è stata rappresentata nelle entità, associazioni, attributi, identificatori, cardinalità dell'E-R o nelle business rules. Verificare la coerenza degli identificatori delle entità e delle sottoentità gerarchiche con i requisiti riscritti.
  - Leggibilità: L'E-R è intuitivo? I nomi dati alle entità/associazioni sono facilmente comprensibili? È chiaro cosa rappresentano?
  - Minimalità: Sono presenti ridondanze indesiderate? È possibile rappresentare le stesse informazioni in modo più semplice
1. Progettazione logica
    1. Compilare una tavola dei volumi (motivare le scelte effettuate)
    2. Compilare una tavola delle operazioni (basandosi anche sui requisiti, includere le operazioni più rilevanti e le scelte effettuate)
    3. Ristrutturazione dello schema E-R
      1. Analisi delle ridondanze Per ogni ridondanza: Per ogni operazione significativa su cui la presenza/assenza della ridondanza può avere effetto:
        - Tavola degli accessi in presenza e in assenza di ridondanza
        - Confronto in spazio e tempo tra presenza e assenza di ridondanza
        - Scelta se introdurre o non introdurre la ridondanza con motivazione
      2. Eliminazione delle generalizzazioni (motivare le scelte effettuate)
      3. Eventuale partizionamento/accorpamento di entità e associazioni (motivare le scelte effettuate)
      4. Eventuale scelta degli identificatori principali (motivare le scelte effettuate)
      5. Schema E-R ristrutturato + regole aziendali
      6. Schema relazionale (indicare anche i vincoli di integrità referenziale)
  2. DDL di creazione del database
  3. DML di popolamento di tutte le tabelle del database (se popolate il database con dati verosimili, potreste rendervi conto di errori commessi nella fase di progettazione concettuale e di cui avreste dovuto rendervi conto prima)
  4. Qualche operazione di cancellazione e modifica per verificare i vincoli e effetti causati da operazioni su chiavi esterne.

## Requisiti iniziali: Piattaforma di live streaming

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 (**guest**) (**l'Utente**

quest viene gestito all'esterno del database). 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 utenti registrati **possono abbonarsi** (diventando utenti "premium") ed avere così accesso a **contenuti extra** come interviste esclusive, podcast e chattare direttamente con i conduttori/artisti protagonisti dei film/concerti etc..

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 ma ad ogni diretta live, viene inviata una notifica agli utenti che seguono il canale **(nell'applicazione, effettuiamo una query al momento dell'avvio di una live da parte dello streamer. Quando lo streamer preme il pulsante 'Inizia Live', la live viene avviata e contemporaneamente eseguiamo una query che inizia dalla tabella 'Live'. Da qui, facciamo una join con la tabella 'Canale', la quale a sua volta si congiunge con la tabella 'Utente'. A questo punto, identifichiamo gli utenti che seguono il canale e inviamo loro una notifica informandoli che la live è iniziata).** Ognuno ha un **titolo**, una **durata**, appartiene a una **categoria** (Figura 1(b)) e **può essere associato** a diversi **hashtags/emojis** etc..). Per ogni live, viene memorizzato il **numero medio di spettatori**, i **commenti** e le **reazioni** (emojis, hashtags etc..) mentre per i video e le clip il **numero di visualizzazioni**.

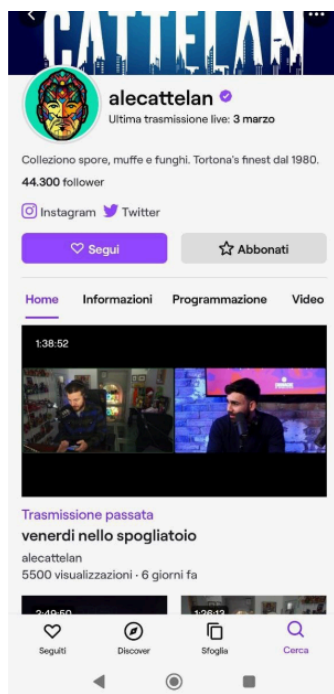


Figura 2(a): La home del canale

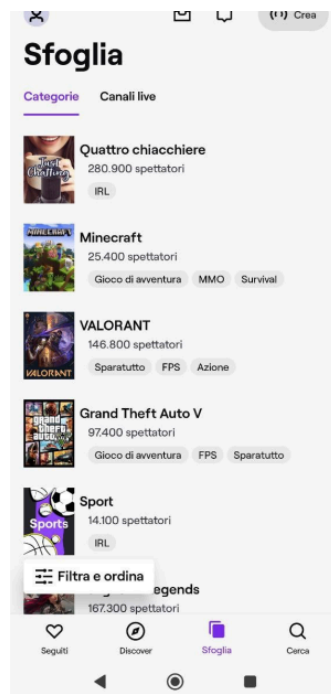


Figura 1(b): La lista delle

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 ogni creatore di contenuti, si

memorizzano il **numero di live effettuate**, il **numero di minuti trasmessi** (in diretta e non) e il **numero medio di spettatori/utenti simultanei** (sia premium che guest). Inoltre, sulla pagina del canale viene visualizzato il numero di followers (**quando un utente visualizza un canale, verrà eseguita una query con una funzione di conteggio (count) che determinerà il numero di utenti che seguono quel canale. Il collegamento tra il 'Canale' e l'Utente' avviene attraverso una join**).

Quando uno streamer rispetta determinati parametri di performance (un minimo di 500 minuti trasmessi, una media di tre o più spettatori simultanei, almeno 50 followers, **quando un canale soddisfa i requisiti necessari, verrà eseguita una query per modificare l'attributo 'Affiliate' impostandolo su 'true'. Abbiamo scelto di utilizzare un booleano per l'attributo 'Affiliate' in quanto non richiede ulteriori valori oltre a essere affiliato o non affiliato**), 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. Inoltre, ogni streamer può anche decidere di trasmettere dirette live solo agli utenti premium (che hanno accesso a contenuti esclusivi).

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 (**gestito dall'applicazione**). I viewer possono inoltre supportare gli streamer tramite la **subscription** (a **pagamento** (**poiché il prezzo di una sottoscrizione potrebbe variare, abbiamo deciso di introdurre un attributo nella relazione di sottoscrizione per gestire questa variazione**)) al loro canale, ottenendo dei privilegi (emoticon personalizzate, nessun limite di caratteri nella lunghezza dei commenti, ecc., **i privilegi degli utenti saranno gestiti in base alla loro adesione o meno alla sottoscrizione**). Inoltre, gli utenti hanno un **portafoglio di bit** (moneta virtuale che possono acquistare tramite la piattaforma), utilizzabile per effettuare donazioni agli streamer tramite differenti metodi di pagamento elettronici (**all'atto del pagamento, l'utente avrà la possibilità di selezionare il metodo di pagamento durante la fase di checkout**).

Una volta che i viewer diventano follower, possono votare i contenuti multimediali degli streamer, esprimendosi tramite l'utilizzo di un voto su scala **likert** (nel range [1,10], **l'attributo è vincolato a valori interi che devono essere compresi tra uno e dieci, inclusi. Questo significa che può assumere solo numeri interi maggiori o uguali a uno e minori o uguali a dieci, garantendo una gamma specifica e limitata di valori**). Oltre a chattare pubblicamente, gli utenti – attraverso un sistema di messaggistica privato embedded nella piattaforma - possono scambiarsi **messaggi** e **contenuti multimediali** privati.

Gli utenti (sia guest che registrati), possono cercare i contenuti multimediali per hashtag o per categorie (ad es. Musica, Sport, Personaggi famosi, Arte, Talk-show, Games, Simulation, food&drinks, Creative, Strategy, Technology, etc...). Ogni utente registrato, in base ai contenuti/pagine/streamer che segue, ha una lista di contenuti multimediali suggeriti (**tutte queste operazioni devono essere gestite attraverso una sequenza di query**

**nell'applicazione. Utilizzando un textbox e un bottone, l'utente può attivare una query che cerca all'interno delle tabelle gli hashtag o categorie desiderate. Nel caso in cui non venga trovato nulla, l'applicazione visualizzerà gli hashtag o categorie di tendenza o quelli di solito seguiti dall'utente).**

Infine, gli **utenti fragili**, possono registrarsi al servizio avendo però a disposizione **contenuti multimediali più inclusivi e accessibili** (ad es. contenuti in LIS – Lingua Italiana dei Segni per le persone non udenti), oppure una versione delle pagine ad accesso facilitato (caratteri aumentati, stile delle pagine dei canali in b/w, **quando un utente è classificato come fragile, l'applicazione si adatterà alle sue specifiche esigenze in base alla sua disabilità, fornendo un'esperienza personalizzata a favore dell'utente**).

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
- Una volta al giorno, viene calcolata la media dei like per ogni contenuto multimediale (per ogni streamer)
- Una volta al giorno, gli amministratori, per ogni contenuto multimediale di ogni streamer, stilano il rating dei video più votati (quelli che, in media, hanno ricevuto una votazione maggiore rispetto agli altri)
- Dieci volte al giorno, vengono controllati ed eliminati tutti i commenti con contenuti offensivi fatti dagli utenti per ogni contenuto multimediale (per ogni streamer), nelle categorie, nei canali e durante le dirette live
- Due volte al giorno vengono controllati i nuovi utenti registrati (sia premium – nuovi abbonati – sia utenti appartenenti a categorie fragili)
- Cinque volte al giorno, gli amministratori delle pagine degli streamer segnalano agli amministratori della base di dati, i profili fake che seguono i loro streamer
- Una volta ogni 6 mesi, gli amministratori possono visualizzare lo storico degli utenti premium (quelli storici (dato un range di date) che quelli recenti (relativi all'ultimo mese))

Qualsiasi altra operazione/funzionalità del sistema e/o modellazione di requisiti non descritti, purché motivata, è ben accetta! (... un po' di fantasia!!!)

Si può assumere che i contenuti multimediali vengano gestiti da una piattaforma/server di video hosting esterna (e che quindi sia sufficiente memorizzare solo un URL o indirizzo IP).

Per il servizio di hosting, gli amministratori delle pagine, devono pagare un corrispettivo mensile di 50\$ al provider che fornisce il servizio di hosting. Se si verificano ritardi per un massimo di 15 giorni a partire dalla data di acquisto/rinnovo dell'hosting, il profilo/canale dello streamer verrà sospeso fino alla data di rinnovo (data di accredito) del pagamento del servizio di hosting.

Legenda:

- **Entità;**
- **Gerarchia;**
- **Attributo;**
- **Relazione;**
- **Implementazione specifiche.**

## 1.2. Glossario dei termini

| Termine | Descrizione  | Sinonimi  | Collegamenti  |
|---------|--|---|---|
| Utente  | La piattaforma consente agli utenti di partecipare a live streaming come spettatori, streamer o entrambi. Gli utenti possono accedere come ospiti o registrarsi per benefici esclusivi. Gli streamer gestiscono canali con live, video e clip, e interagiscono con la community attraverso chat, follow e abbonamenti. I follower possono votare contenuti, cercare per hashtag o categorie e ricevere suggerimenti personalizzati. Per gli utenti fragili, sono disponibili opzioni di accesso facilitato. L'ispirazione a Twitch è evidente nella gestione dettagliata dei dati e nei requisiti per diventare affiliati. | User, viewer, streamer, utente fragile, guest, follower | Messaggio privato, canale, live, contenuto, reazione, commento, sub |
| Admin   | Gli amministratori gestiscono qualifica e like, calcolano la classifica settimanale. Controllano commenti, nuovi utenti e profili fake. Mensilmente gestiscono pagamenti hosting, sospensioni in caso di ritardi. Ogni 6 mesi visualizzano storico utenti premium.   |   | Canale  |
| Canale  | Il canale sulla piattaforma è il luogo gestito dagli streamer per condividere live, video e clip. Interagisce con la community attraverso chat, follow e abbonamenti. I follower possono partecipare alle live, votare contenuti e cercare hashtag. Il canale è collegato ai social, offrendo una piattaforma completa per esplorare e condividere contenuti.  |   | Social, contenuto, live, utente, admin, sub                         |
| Sub     | La sub è una donazione che fa un utente ad un canale per supportarlo.  | Subscription  | Utente, canale  |
| Social  | I social sono collegati al canale, offrendo una piattaforma completa per esplorare e condividere contenuti.  |   | Canale  |

| <b>Termine</b>    | <b>Descrizione</b>   | <b>Sinonimi</b> | <b>Collegamenti</b>                |
|-------------------|--|-----------------|------------------------------------|
| Contenuto         | Il contenuto sulla piattaforma può assumere diverse forme, tra cui clip e video. I contenuti sono gestiti sul canale dello streamer. I follower votano i contenuti e li cercano attraverso hashtag o emoji. Una live può essere pubblicata come video sul canale. L'intera esperienza di condivisione di contenuti è centralizzata attraverso i canali, contribuendo a creare una piattaforma dinamica e coinvolgente. | Clip, video     | Canale, live, utente               |
| Live              | Le live sulla piattaforma, che possono essere esclusive, rappresentano un momento in cui gli utenti partecipano in tempo reale, interagendo attraverso commenti e reazioni. L'utente si può connettere alla live direttamente dal canale.  | Live esclusiva  | Commento, reazione, utente, canale |
| Reazione          | Le reazioni sulla piattaforma, rappresentate da hashtag ed emoji, offrono agli utenti la possibilità di esprimere risposte durante una live. Questi elementi arricchiscono l'interazione visiva e simbolica, collegando gli utenti al flusso dinamico di contenuti e alle interazioni sociali durante la trasmissione in tempo reale.  | Hashtag, emoji  | Live, utente                       |
| Commento          | I commenti sulla piattaforma permettono agli utenti di esprimere direttamente le loro opinioni durante una live. Questa funzione collega gli utenti direttamente alla trasmissione in corso, rendendo l'esperienza più coinvolgente e socialmente interattiva.   |                 | Live, utente                       |
| Messaggio privato | I messaggi sulla piattaforma consentono agli utenti di interagire in modo privato tra loro. Gli utenti possono scambiarsi messaggi, creando connessioni personali e facilitando la comunicazione privata. La funzione di messaggistica rafforza i legami tra gli utenti, offrendo un canale di comunicazione diretta e riservata.  |                 | Utente                             |
| Abbonamento       | Gli utenti registrati possono abbonarsi (diventando utenti "premium") ed avere così accesso a contenuti extra come interviste esclusive, podcast e chattare direttamente con i conduttori/artisti protagonisti dei film/concerti etc.  |                 | Utente                             |

### 1.3. Requisiti riscritti

~~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~~ **Gli utenti** possono essere registrati al servizio oppure possono guardare le live in modo anonimo (~~guest~~) (~~quando un utente accede al sito, gli verranno presentate tre opzioni: effettuare il login, registrarsi (sign up) o accedere come ospite (guest).~~ Se un utente sceglie di accedere come ~~ospite~~ **utente anonimo**, non avrà accesso a tutti i vantaggi riservati agli utenti registrati. ~~Poiché la sua presenza è anonima e non è associata ad un nickname, indirizzo email o numero di telefono, abbiamo deciso di utilizzare un ID come chiave primaria per l'entità 'Utente'.~~ Per registrarsi, gli utenti devono indicare nome utente, password, data di nascita, numero di telefono o indirizzo mail. Gli utenti ~~iseriti~~ **registrati** possono chattare, seguire ~~lo streamer~~ **creatori di contenuti**, creare dirette. Gli utenti registrati possono abbonarsi, **potendo così avere ulteriori vantaggi quali contenuti extra e chattare** (~~diventando utenti "premium"~~) ed avere ~~così accesso a contenuti extra come interviste esclusive, podcast e chattare direttamente con i conduttori/artisti protagonisti dei film/concerti etc..~~

Gli **utenti possono avere** ~~streamer hanno ciascuno~~ un canale, che può essere caratterizzato tramite una descrizione. Al ~~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 ma ad ogni diretta live, viene inviata una notifica agli utenti che seguono il canale (~~nell'applicazione, effettuiamo una query al momento dell'avvio di una live da parte dello streamer. Quando lo streamer preme il pulsante 'Inizia Live', la live viene avviata e contemporaneamente eseguiamo una query che inizia dalla tabella 'Live'. Da qui, facciamo una join con la tabella 'Canale', la quale a sua volta si congiunge con la tabella 'Utente'. A questo punto, identifichiamo gli utenti che seguono il canale e inviamo loro una notifica informandoli che la live è iniziata~~). Ognuno ha un titolo, una durata, appartiene a una categoria (**Figura 1(b)**) e può essere associato a diversi hashtags/emojis etc..). Per ogni live, viene memorizzato il numero medio di spettatori, i commenti e le reazioni (emojis, hashtags etc..) mentre per i video e le clip il numero di visualizzazioni.

~~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 ogni **creatore di utente che crea** contenuti, si memorizzano il numero di live effettuate, il numero di minuti trasmessi (in diretta e non) e il numero medio di ~~spettatori~~ utenti simultanei (sia premium che guest). Inoltre, sulla pagina del canale viene visualizzato il numero di followers (~~quando un utente visualizza un canale, verrà eseguita una query con una funzione di conteggio (count) che~~



~~determinerà il numero di utenti che seguono quel canale. Il collegamento tra il 'Canale' e l'Utente avviene attraverso una join).~~

Quando uno ~~streamer~~ **utente** rispetta determinati parametri di performance (un minimo di 500 minuti trasmessi, una media di tre o più spettatori simultanei, almeno 50 followers, ~~quando un canale soddisfa i requisiti necessari, verrà eseguita una query per modificare l'attributo 'Affiliate' impostandolo su 'true'. Abbiamo scelto di utilizzare un booleano per l'attributo 'Affiliate' in quanto non richiede ulteriori valori oltre a essere affiliato o non affiliato~~), può diventare affiliate. Le **stream live** hanno degli orari. Ogni **streamer utente che crea contenuti** ha un calendario in cui può dire quando farà stream e indicare il titolo delle prossime live. Inoltre, ~~ogni streamer~~ può anche decidere di trasmettere dirette live solo agli utenti premium (che hanno accesso a contenuti esclusivi).

~~I viewer~~ **Gli utenti registrati** possono diventare follower del canale degli **streamer utenti che creano contenuti** che preferiscono, e le loro preferenze sono raccolte in un elenco di follower a cui possono accedere dal loro profilo (~~gestito dall'applicazione~~). ~~I viewer~~ **Gli utenti registrati** possono inoltre supportare gli **streamer utenti che creano contenuti** tramite la subscription (a pagamento (~~poiché il prezzo di una sottoscrizione potrebbe variare, abbiamo deciso di introdurre un attributo nella relazione di sottoscrizione per gestire questa variazione~~)) al loro canale, ottenendo dei privilegi (emoticon personalizzate, nessun limite di caratteri nella lunghezza dei commenti, ecc., ~~i privilegi degli utenti saranno gestiti in base alla loro adesione o meno alla sottoscrizione~~). Inoltre, gli utenti hanno un portafoglio di bit (moneta virtuale che possono acquistare tramite la piattaforma), utilizzabile per effettuare donazioni agli **streamer utenti che creano contenuti** tramite differenti metodi di pagamento elettronici (~~all'atto del pagamento, l'utente avrà la possibilità di selezionare il metodo di pagamento durante la fase di checkout~~).

Una volta che ~~i viewer~~ **gli utenti registrati** diventano follower, possono votare i contenuti multimediali degli **streamer utenti che creano contenuti**, esprimendosi tramite l'utilizzo di un voto su scala likert (nel range [1,10], ~~l'attributo è vincolato a valori interi che devono essere compresi tra uno e dieci, inclusi. Questo significa che può assumere solo numeri interi maggiori o uguali a uno e minori o uguali a dieci, garantendo una gamma specifica e limitata di valori~~). Oltre a chattare pubblicamente, gli utenti – attraverso un sistema di messaggistica privato embedded nella piattaforma - possono scambiarsi messaggi e contenuti multimediali privati.

Gli utenti (sia guest che registrati), possono cercare i contenuti multimediali per hashtag o per categorie (ad es. Musica, Sport, Personaggi famosi, Arte, Talk-show, Games, Simulation, food&drinks, Creative, Strategy, Technology, etc...). Ogni utente registrato, in base ai contenuti/pagine/**streamer utente che crea contenuti** che segue, ha una lista di contenuti multimediali suggeriti (~~tutte queste operazioni devono essere gestite attraverso una sequenza di query nell'applicazione. Utilizzando un textbox e un bottone, l'utente può attivare una query che cerca all'interno delle tabelle gli hashtag o categorie desiderate. Nel caso in cui non~~

~~venga trovato nulla, l'applicazione visualizzerà gli hashtag o categorie di tendenza o quelli di solito seguiti dall'utente).~~

Infine, gli utenti fragili, possono registrarsi al servizio avendo però a disposizione contenuti multimediali più inclusivi e accessibili (ad es. contenuti in LIS – Lingua Italiana dei Segni per le persone non udenti), oppure una versione delle pagine ad accesso facilitato (caratteri aumentati, stile delle pagine dei canali in b/w, ~~quando un utente è classificato come fragile, l'applicazione si adatterà alle sue specifiche esigenze in base alla sua disabilità, fornendo un'esperienza personalizzata a favore dell'utente).~~

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
- Una volta al giorno, viene calcolata la media dei like per ogni contenuto multimediale (per ogni streamer)
- Una volta al giorno, gli amministratori, per ogni contenuto multimediale di ogni streamer, stilano il rating dei video più votati (quelli che, in media, hanno ricevuto una votazione maggiore rispetto agli altri)
- Dieci volte al giorno, vengono controllati ed eliminati tutti i commenti con contenuti offensivi fatti dagli utenti per ogni contenuto multimediale (per ogni streamer), nelle categorie, nei canali e durante le dirette live
- Due volte al giorno vengono controllati i nuovi utenti registrati (sia premium – nuovi abbonati – sia utenti appartenenti a categorie fragili)
- Cinque volte al giorno, gli amministratori delle pagine degli streamer segnalano agli amministratori della base di dati, i profili fake che seguono i loro streamer
- Una volta ogni 6 mesi, gli amministratori possono visualizzare lo storico degli utenti premium (quelli storici (dato un range di date) che quelli recenti (relativi all'ultimo mese)

~~Qualsiasi altra operazione/funzionalità del sistema e/o modellazione di requisiti non descritti, purché motivata, è ben accetta! (... un po' di fantasia!!!)~~

~~Si può assumere che i contenuti multimediali vengano gestiti da una piattaforma/server di video hosting esterna (e che quindi sia sufficiente memorizzare solo un URL o indirizzo IP).~~

Per il servizio di hosting, gli **amministratori admin** delle pagine, devono pagare un corrispettivo mensile di 50\$ al provider che fornisce il servizio di hosting. Se si verificano ritardi per un massimo di 15 giorni a partire dalla data di acquisto/rinnovo dell'hosting, il profilo/canale dello **streamer creatore di contenuti** verrà sospeso fino alla data di rinnovo (data di accredito) del pagamento del servizio di hosting.

< qui ci vanno i requisiti della sezione 1.1 riscritti senza sinonimi e con frasi standardizzate. Ricordarsi di marcare in maniera grafica ~~ciò che è stato cancellato/sostituito~~ e **ciò che è stato aggiunto**>

## 1.4. Requisiti strutturati in gruppi di frasi omogenee

### Frasi di carattere generale

Gli utenti possono essere registrati al servizio oppure possono guardare le live in modo anonimo. Se un utente sceglie di accedere come utente anonimo, non avrà accesso a tutti i vantaggi riservati agli utenti registrati. Gli utenti registrati possono chattare, seguire creatori di contenuti, creare dirette, abbonarsi, potendo così avere ulteriori vantaggi quali contenuti extra e chattare.

Gli utenti possono avere un canale. Al canale, è possibile specificare una lista di social associati, un'immagine profilo e anche un trailer. In ogni canale possono esserci live, video (live passate) e clip (video di durata breve). Ognuno ha un titolo, una durata, appartiene a una categoria e può essere associato a diversi hashtags/emojis etc. In ogni live, si possono fare commenti e reazioni.

Sulla pagina del canale viene visualizzato il numero di followers.

Quando un utente rispetta determinati parametri di performance, può diventare affiliate. Le live hanno degli orari. Ogni utente che crea contenuti ha un calendario in cui può dire quando farà stream e indicare il titolo delle prossime live. Inoltre può anche decidere di trasmettere dirette live solo agli utenti premium (che hanno accesso a contenuti esclusivi).

Gli utenti registrati possono diventare follower del canale degli utenti che creano contenuti che preferiscono, e le loro preferenze sono raccolte in un elenco di follower a cui possono accedere dal loro profilo. Gli utenti registrati possono inoltre supportare gli utenti che creano contenuti tramite la subscription (a pagamento) al loro canale, ottenendo dei privilegi. Inoltre, gli utenti hanno un portafoglio di bit, utilizzabile per effettuare donazioni agli utenti che creano contenuti tramite differenti metodi di pagamento elettronici.

Una volta che gli utenti registrati diventano follower, possono votare i contenuti multimediali degli utenti che creano contenuti. Oltre a chattare pubblicamente, gli utenti possono scambiarsi messaggi e contenuti multimediali privati.

Gli utenti (sia guest che registrati), possono cercare i contenuti multimediali per hashtag o per categorie. Ogni utente registrato, in base ai contenuti/pagine/utente che crea contenuti che segue, ha una lista di contenuti multimediali suggeriti .

Infine, gli utenti fragili, possono registrarsi al servizio avendo però a disposizione contenuti multimediali più inclusivi e accessibili, oppure una versione delle pagine ad accesso facilitato (caratteri aumentati, stile delle pagine dei canali in b/w).

Per il servizio di hosting, gli admin delle pagine, devono pagare un corrispettivo mensile di 50\$ al provider che fornisce il servizio di hosting. Se si verificano ritardi per un massimo di 15 giorni a partire dalla data di acquisto/rinnovo dell'hosting, il profilo/canale del creatore di contenuti verrà sospeso fino alla data di rinnovo (data di accredito) del pagamento del servizio di hosting.

#### Fraasi relative a Utente

Gli utenti possono essere registrati al servizio oppure possono guardare le live in modo anonimo. Se un utente sceglie di accedere come utente anonimo, non avrà accesso a tutti i vantaggi riservati agli utenti registrati. Per registrarsi, gli utenti devono indicare nome utente, password, data di nascita, numero di telefono o indirizzo mail. Gli utenti registrati possono chattare, seguire creatori di contenuti, creare dirette. Gli utenti registrati possono abbonarsi, potendo così avere ulteriori vantaggi quali contenuti extra e chattare.

Gli utenti possono avere un canale, che può essere caratterizzato tramite una descrizione. Al canale, è possibile specificare una lista di social associati (ad esempio Instagram, YouTube, ecc.), un'immagine profilo e anche un trailer.

Per ogni utente che crea contenuti, si memorizzano il numero di live effettuate, il numero di minuti trasmessi (in diretta e non) e il numero medio di utenti simultanei (sia premium che guest). Inoltre, sulla pagina del canale viene visualizzato il numero di followers.

Quando un utente rispetta determinati parametri di performance (un minimo di 500 minuti trasmessi, una media di tre o più spettatori simultanei, almeno 50 followers), può diventare affiliate. Ogni utente che crea contenuti ha un calendario in cui può dire quando farà stream e indicare il titolo delle prossime live. Inoltre può anche decidere di trasmettere dirette live solo agli utenti premium (che hanno accesso a contenuti esclusivi).

Gli utenti registrati possono diventare follower del canale degli utenti che creano contenuti che preferiscono, e le loro preferenze sono raccolte in un elenco di follower a cui possono accedere dal loro profilo. Gli utenti registrati possono inoltre supportare gli utenti che creano contenuti tramite la subscription (a pagamento) al loro canale, ottenendo dei privilegi (emoticon personalizzate, nessun limite di caratteri nella lunghezza dei commenti, ecc.). Inoltre, gli utenti hanno un portafoglio di bit (moneta virtuale che possono acquistare tramite la piattaforma), utilizzabile per effettuare donazioni agli utenti che creano contenuti tramite differenti metodi di pagamento elettronici.

Una volta che gli utenti registrati diventano follower, possono votare i contenuti multimediali degli utenti che creano contenuti, esprimendosi tramite l'utilizzo di un voto su scala likert (nel range [1,10]). Oltre a chattare pubblicamente, gli utenti – attraverso un sistema di messaggistica privato embedded nella piattaforma - possono scambiarsi messaggi e contenuti multimediali privati.

Gli utenti (sia guest che registrati), possono cercare i contenuti multimediali per hashtag o per categorie (ad es. Musica, Sport, Personaggi famosi, Arte, Talk-show, Games, Simulation, food&drinks, Creative, Strategy, Technology, etc...). Ogni utente registrato, in base ai contenuti/pagine/utente che crea contenuti che segue, ha una lista di contenuti multimediali suggeriti.

Infine, gli utenti fragili, possono registrarsi al servizio avendo però a disposizione contenuti multimediali più inclusivi e accessibili (ad es. contenuti in LIS – Lingua Italiana dei Segni per le persone non udenti), oppure una versione delle pagine ad accesso facilitato (caratteri aumentati, stile delle pagine dei canali in b/w).

#### Frase relative a Admin

Per il servizio di hosting, gli admin delle pagine, devono pagare un corrispettivo mensile di 50\$ al provider che fornisce il servizio di hosting. Se si verificano ritardi per un massimo di 15 giorni a partire dalla data di acquisto/rinnovo dell'hosting, il profilo/canale del creatore di contenuti verrà sospeso fino alla data di rinnovo (data di accredito) del pagamento del servizio di hosting.

#### Frase relative a Sub

Gli utenti registrati possono inoltre supportare gli utenti che creano contenuti tramite la subscription (a pagamento) al loro canale, ottenendo dei privilegi (emoticon personalizzate, nessun limite di caratteri nella lunghezza dei commenti, ecc.).

#### Frase relative a Social

Al canale, è possibile specificare una lista di social associati (ad esempio Instagram, YouTube, ecc.).

#### Frase relative a Contenuto

In ogni canale possono esserci live, video (live passate) e clip (video di durata breve).

Una volta che gli utenti registrati diventano follower, possono votare i contenuti multimediali degli utenti che creano contenuti, esprimendosi tramite l'utilizzo di un voto su scala likert (nel range [1,10]).

#### Frase relative a Canale

Gli utenti possono avere un canale, che può essere caratterizzato tramite una descrizione. Al

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 ma ad ogni diretta live, viene inviata una notifica agli utenti che seguono il canale.

#### Frase relative a Live

Le live possono anche non diventare video del canale ma ad ogni diretta live, viene inviata una notifica agli utenti che seguono il canale. Ognuno ha un titolo, una durata, appartiene a una categoria e può essere associato a diversi hashtags/emojis etc.). Per ogni live, viene memorizzato il numero medio di spettatori, i commenti e le reazioni (emojis, hashtags etc..) mentre per i video e le clip il numero di visualizzazioni.

Le live hanno degli orari. Ogni utente che crea contenuti ha un calendario in cui può dire quando farà stream e indicare il titolo delle prossime live. Inoltre può anche decidere di trasmettere dirette live solo agli utenti premium (che hanno accesso a contenuti esclusivi).

#### Frase relative a Reazione

Per ogni live, viene memorizzato il numero medio di spettatori, i commenti e le reazioni (emojis, hashtags etc..) mentre per i video e le clip il numero di visualizzazioni.

#### Frase relative a Commento

Per ogni live, viene memorizzato il numero medio di spettatori, i commenti e le reazioni (emojis, hashtags etc..) mentre per i video e le clip il numero di visualizzazioni.

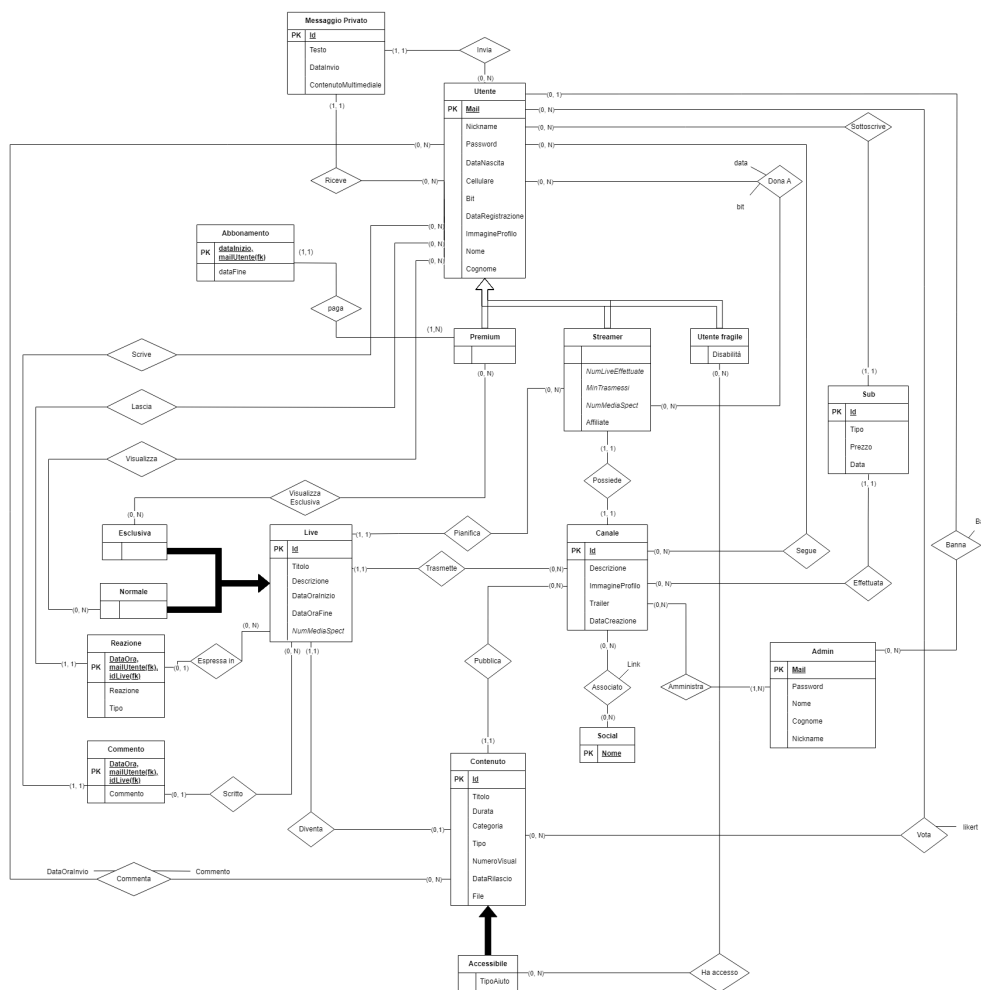
#### Frase relative a Messaggio privato

Oltre a chattare pubblicamente, gli utenti – attraverso un sistema di messaggistica privato embedded nella piattaforma - possono scambiarsi messaggi e contenuti multimediali privati.

#### Frase relative a Abbonamento

Gli utenti registrati possono abbonarsi (diventando utenti “premium”) ed avere così accesso a contenuti extra come interviste esclusive, podcast e chattare direttamente con i conduttori/artisti protagonisti dei film/concerti etc.

## 1.5. Schema E-R + regole aziendali



La gerarchia degli utenti, composta da Premium, Streamer e Utente Fragile, è una gerarchia parziale sovrapposta. Non è obbligatorio che un utente sia uno streamer, fragile o premium, permettendo una flessibilità nella classificazione. Inizialmente consideravamo un figlio Viewer, ma essendo tutti utenti, è stato escluso, semplificando la struttura e unendo direttamente le caratteristiche di visualizzazione all'entità utente principale. Questa modifica migliora la chiarezza e la coerenza della gerarchia.

La gerarchia dei tipi di Live, suddivisa tra Esclusiva e Normale, è di tipo congiuntivo. Questa divisione permette di differenziare chiaramente le due tipologie di Live, indicando in modo esplicito che solo gli utenti abbonati hanno accesso alle Live esclusive. In una gerarchia di tipo congiuntivo, ogni elemento deve appartenere ad almeno una delle categorie specificate. In questo caso, ogni tipo di Live è chiaramente definito e associato a una condizione specifica, migliorando la comprensione delle restrizioni di accesso.

La gerarchia di Contenuto con Accessibile è di tipo disgiuntiva. Abbiamo deciso di creare una gerarchica per far capire la netta distinzione tra un Contenuto Accessibile con Utente fragile e un Utente senza disabilità con i contenuti accessibili a tutti.

Abbiamo deciso di introdurre l'entità sub perché esistono diversi tipi di sottoscrizione ad un Canale. Inoltre ogni tipo di sottoscrizione ha un costo diverso.

Non abbiamo creato due entità separate per distinguere se una Reazione è un hashtag o un emoji ma ne abbiamo creata solamente una che li differenzia grazie all'attributo tipo.

Il pattern di progettazione utilizzato è il modello gerarchico. Nel corso del nostro progetto, abbiamo adottato un pattern di progettazione chiave che ha svolto un ruolo fondamentale nella definizione della struttura del nostro sistema. In particolare, abbiamo implementato il modello gerarchico come approccio principale alla progettazione del nostro modello Entità-Relazione (ER).

La nostra strategia di progetto ha iniziato con un'analisi attenta del testo per identificare le entità primarie, tra cui Utente e Canale. Abbiamo successivamente individuato gerarchie chiave per differenziarle, consentendo una struttura chiara nel modello Entità-Relazione (E-R). Una volta identificate le entità, ci siamo concentrati sulla ricerca degli attributi pertinenti per ciascuna di esse. Durante la fase di realizzazione dell'E-R, abbiamo stabilito collegamenti tra le entità mediante associazioni ben definite. Questo approccio ci ha permesso di costruire una rappresentazione solida e completa delle relazioni e delle caratteristiche chiave nel nostro sistema.

## Regole aziendali

### Vincoli di Integrità:

|      |  |
|------|--|
| RV1  | Messaggio Privato deve avere un solo mittente e un solo destinatario |
| RV2  | Live esclusiva deve essere vista solo da utenti premium              |
| RV3  | Live deve essere relativa ad un canale                               |
| RV4  | Live deve essere pianificata da uno streamer                         |
| RV5  | Sub deve essere sottoscritta da un utente                            |
| RV6  | Sub deve essere effettuata presso un canale                          |
| RV7  | Contenuto deve essere relativo ad un canale                          |
| RV8  | Live può essere esclusiva  |
| RV9  | Utente può essere premium  |
| RV10 | Utente può essere streamer   |
| RV11 | Utente può essere fragile  |
| RV12 | Utente può essere premium e streamer                                 |



|      |   |
|------|---|
| RV13 | Utente può essere premium e fragile           |
| RV14 | Utente può essere streamer e fragile          |
| RV15 | Utente può essere streamer, premium e fragile |
| RV16 | Contenuto può essere accessibile              |
| RV17 | Contenuto può essere video                    |
| RV18 | Contenuto può essere clip                     |

### Derivazioni:

|     |  |
|-----|--|
| RD1 | Il numero medio di spettatori si ottiene facendo la media di spettatori in live  |
| RD2 | Il numero di live trasmesse da utente si ottiene facendo la somma di tutte le live trasmesse dall'utente                           |
| RD3 | Il numero dei minuti trasmessi da utente si ottiene facendo la somma di tutti i minuti trasmessi in live dall'utente               |
| RD4 | Il numero di spettatori medi presenti si ottiene facendo la media del numero di persone che partecipano in live ogni minuto (live) |
| RD5 | Il numero di spettatori medi presenti si ottiene facendo la media del numero di persone che partecipano in tutte le live (utente)  |
| RD6 | L'affiliate di un utente si ottiene verificando i requisiti: numMediaSpect > 3, followers > 50 e numMinTrasmessi > 500             |
| RD7 | Il numero di visualizzazioni di un contenuto si ottiene facendo la somma di tutti gli utenti che hanno visto quel contenuto        |

## 2 PROGETTAZIONE LOGICA

### 2.1. Tavola dei volumi

| Concetto          | Tipo   | Volume            |
|-------------------|--------|-------------------|
| Utente            | Entità | 15.000.000        |
| Admin             | Entità | 5.000.000         |
| Canale            | Entità | 8.000.000         |
| Sub               | Entità | 500.000.000       |
| Social            | Entità | 10                |
| Contenuto         | Entità | 3.200.000.000.000 |
| Live              | Entità | 16.000.000.000    |
| Reazione          | Entità | 225.000.000.000   |
| Commento          | Entità | 15.000.000        |
| Messaggio privato | Entità | 7.500.000.000.000 |
| Abbonamento       | Entità | 3.000.000         |

|                      |              |                   |
|----------------------|--------------|-------------------|
| Invia                | Associazione | 3.750.000.000.000 |
| Riceve               | Associazione | 3.750.000.000.000 |
| Dona a               | Associazione | 750.000.000       |
| Scrive               | Associazione | 15.000.000        |
| Lascia               | Associazione | 225.000.000.000   |
| Visualizza           | Associazione | 150.000.000.000   |
| Espressa in          | Associazione | 225.000.000.000   |
| Scritto              | Associazione | 15.000.000        |
| Pianifica            | Associazione | 200.000.000       |
| Trasmette            | Associazione | 16.000.000.000    |
| Diventa              | Associazione | 10.000.000.000    |
| Pubblica             | Associazione | 3.200.000.000.000 |
| Associato            | Associazione | 80.000.000        |
| Possiede             | Associazione | 8.000.000         |
| Segue                | Associazione | 1.500.000.000     |
| Sottoscrive          | Associazione | 3.000.000         |
| Paga                 | Associazione | 500.000.000       |
| Vota                 | Associazione | 3.200.000.000.000 |
| Visualizza esclusiva | Associazione | 100.000           |
| Ha accesso           | Associazione | 15.000            |
| CommentaContenuto    | Associazione | 4.500.000.000.000 |

Per quanto riguarda gli utenti (sia viewer che streamer), abbiamo approfondito le nostre ricerche cercando quanti utenti avesse twitch; per tutto il resto abbiamo deciso di fare delle stime e delle medie di quanti utenti potessero eseguire determinate azioni.

## 2.2. Tavola delle operazioni

| Operazione | Descrizione  | Tipo | Frequenza           |
|------------|--|------|---------------------|
| Op1        | Controllare le condizioni per la qualifica di affiliate                        | B    | 1 volta al giorno   |
| Op2        | Calcolare la classifica degli streamer più seguiti                             | B    | 1 volta a settimana |
| Op3        | Calcolare la media dei like per ogni contenuto multimediale(per ogni streamer) | B    | 1 volta al giorno   |

| Operazione | Descrizione   | Tipo | Frequenza          |
|------------|---|------|--------------------|
| Op4        | Gli amministratori, per ogni contenuto multimediale di ogni streamer, stilano il rating dei video più votati (quelli che, in media, hanno ricevuto una votazione maggiore rispetto agli altri)  | B    | 1 volta al giorno  |
| Op5        | Vengono controllati ed eliminati tutti i commenti con contenuti offensivi fatti dagli utenti per ogni contenuto multimediale (per ogni streamer), nelle categorie, nei canali e durante le dirette live   | I    | 10 volte al giorno |
| Op6        | Vengono controllati i nuovi utenti registrati (sia premium – nuovi abbonati – sia utenti appartenenti a categorie fragili)  | B    | 2 volte al giorno  |
| Op7        | Gli amministratori delle pagine degli streamer segnalano agli amministratori della base di dati, i profili fake che seguono i loro streamer   | I    | 5 volte al giorno  |
| Op8        | Gli amministratori possono visualizzare lo storico degli utenti premium (quelli storici (dato un range di date)) che quelli recenti (relativi all'ultimo mese)  | I    | Ogni 6 mesi        |
| Op9        | Calcolare il numero medio di spettatori durante il corso della live facendo la somma tra gli spettatori vecchi (5 minuti prima), il numero di spettatori attuali e dividere il tutto per due. Per i primi 5 minuti, il primo valore sarà sempre per forza 0 | B    | Ogni 5 minuti      |
| Op10       | Fare la medie delle medie dei numeri di spettatori di ogni live effettuata dallo streamer   | B    | 1 volta al giorno  |
| Op11       | Calcolare il numero delle live effettuate dallo streamer  | I    | 1 volta al giorno  |
| Op12       | Calcolare il numero di minuti totali delle live effettuate dallo streamer   | I    | 1 volta al giorno  |

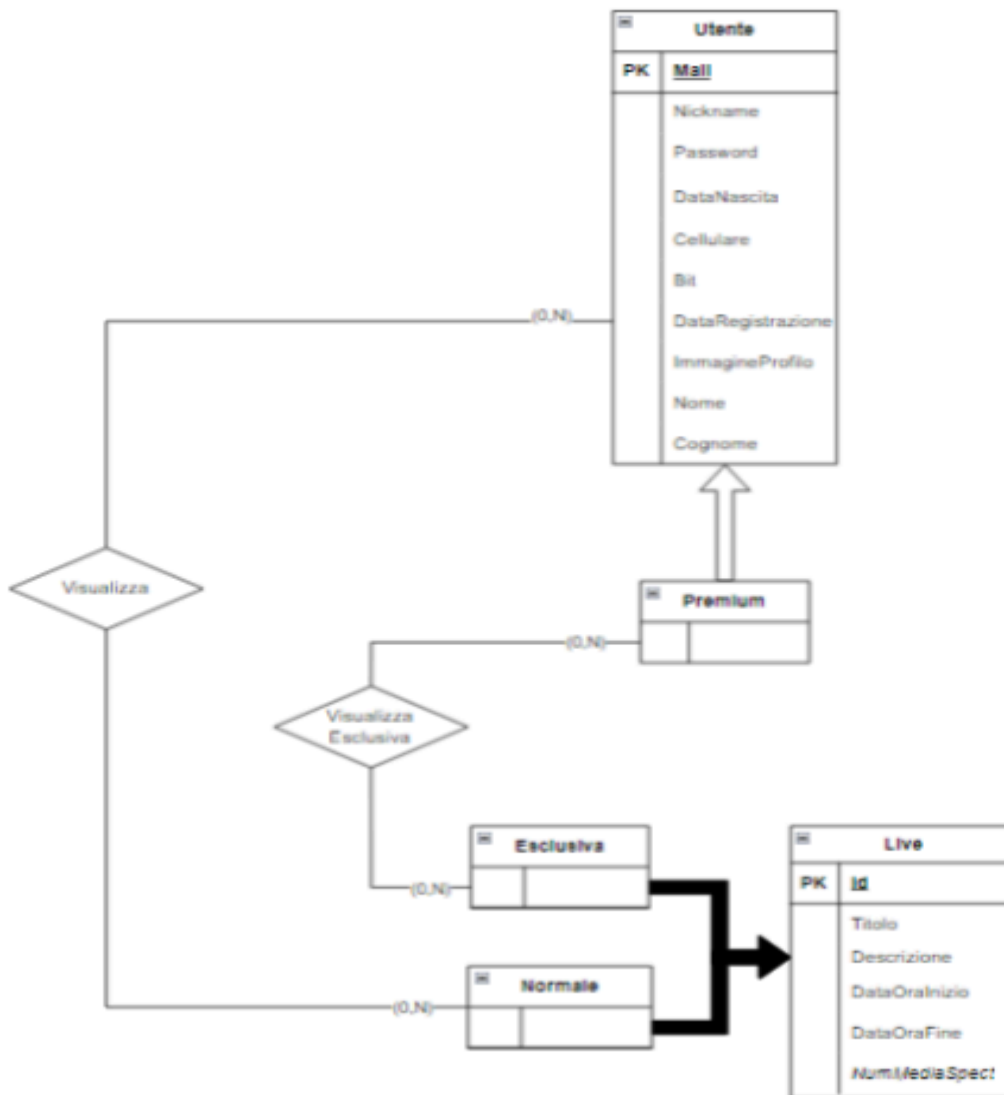
## 2.3. Ristrutturazione dello schema E-R

### 2.3.1. Analisi delle ridondanze

Nel nostro E-R abbiamo individuato 4 ridondanze di tipo:

- da attributi di altre entità (o associazioni);
- dal conteggio delle partecipazioni ad una associazione.

### 2.3.1.1. numeroMediaSpect Live(attributo ricavabile da associazione)



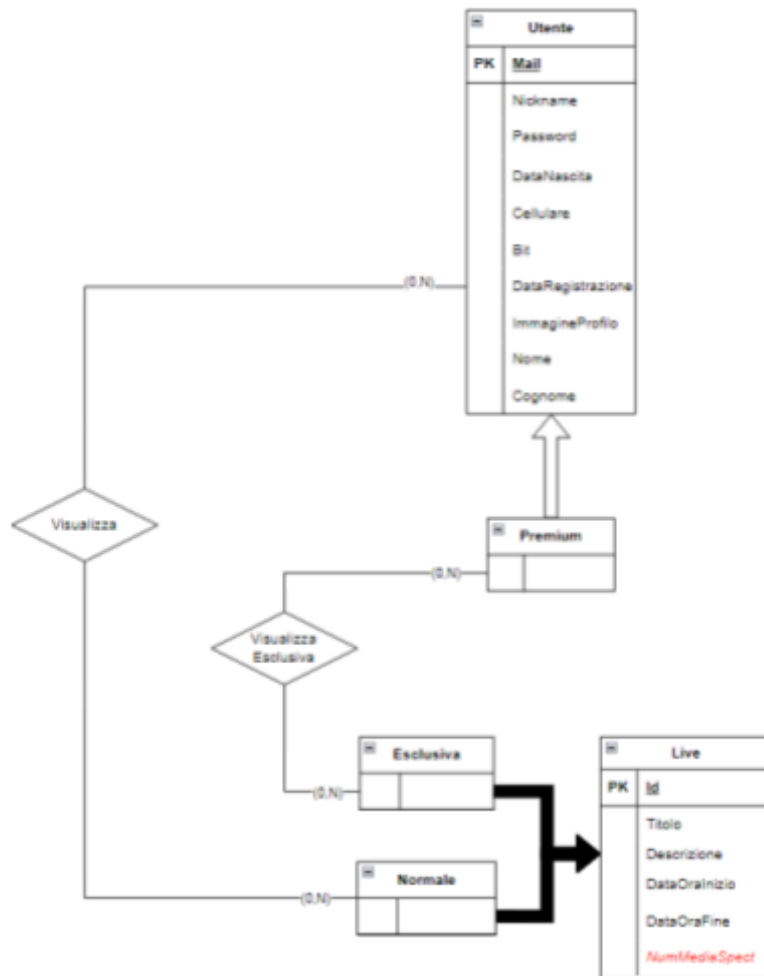
**DERIVAZIONE:** L'attributo è derivabile dal calcolo del numero di visualizzazioni o visualizzazioni esclusive(in base al tipo di live) ad ogni 5 minuti della live

#### OPERAZIONI COINVOLTE

**Op9:** Calcolare il numero medio di spettatori durante il corso della live facendo la somma tra gli spettatori vecchi (5 minuti prima), il numero di spettatori attuali e dividere il tutto per due. Per i primi 5 minuti, il primo valore sarà sempre per forza 0.

#### PRESENZA DI RIDONDANZA

**Op9:** Calcolare il numero medio di spettatori durante il corso della live facendo la somma tra gli spettatori vecchi (5 minuti prima), il numero di spettatori attuali e dividere il tutto per due. Per i primi 5 minuti, il primo valore sarà sempre per forza 0.



| Concetto             | Costrutto    | Accessi | Tipo | Descrizione   |
|----------------------|--------------|---------|------|---|
| Utente               | Entità       | 5       | L    | Mi serve per contare quanti utenti sono spettatori della live |
| Visualizza esclusiva | Associazione | 5       | L    | Eseguo una join per collegare l'entità Utente all'entità Live |
| Visualizza           | Associazione | 5       | L    | Eseguo una join per collegare l'entità Utente all'entità Live |
| Live                 | Entità       | 5       | S    | Scrivo il risultato della query nell'attributo NumMediaSpect  |

Tavola degli accessi:

S: 2 (entità) \* 1 (S) \* 5 (frequenza) = 10 accessi/5 min

L: 2 (associazione) \* 3 (L) \* 5 (frequenza) = 30 accessi/5 min

TOT: 10 + 30 = 40 accessi/5 min

## ASSENZA DI RIDONDANZA

**Op9:** Calcolare il numero medio di spettatori durante il corso della live facendo la somma tra gli spettatori vecchi (5 minuti prima), il numero di spettatori attuali e dividere il tutto per due. Per i primi 5 minuti, il primo valore sarà sempre per forza 0.

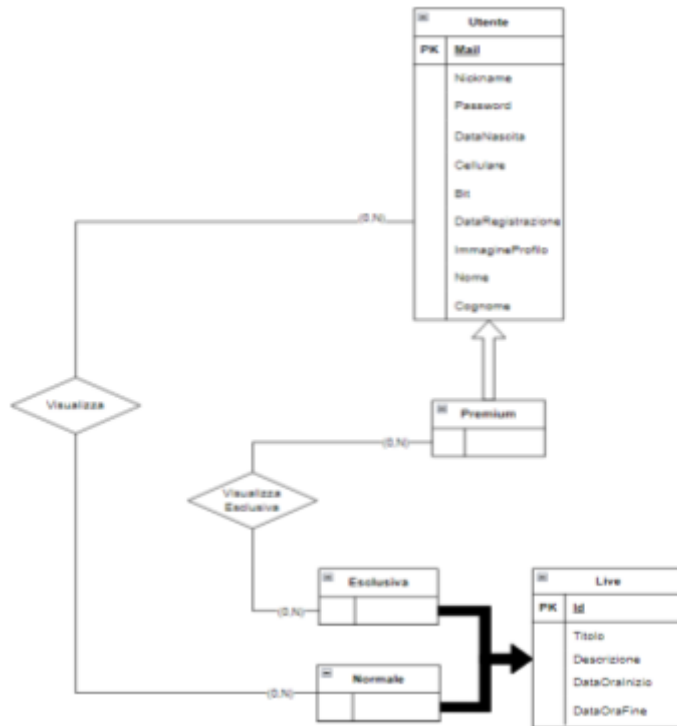


Tavola degli accessi:

| Concetto             | Costrutto    | Accessi | Tipo | Descrizione   |
|----------------------|--------------|---------|------|---|
| Utente               | Entità       | 5       | L    | Mi serve per contare quanti utenti sono spettatori della live |
| Visualizza esclusiva | Associazione | 5       | L    | Eseguo una join per collegare l'entità Utente all'entità Live |
| Visualizza           | Associazione | 5       | L    | Eseguo una join per collegare l'entità Utente all'entità Live |
| Live                 | Entità       | 5       | L    | Mostro a video il risultato della query                       |

S: 0

L: 2 (associazione) \* 4 (L) \* 5 (frequenza) = 40 accessi/5 min

TOT: 40 accessi/5 min

## TOTALI PER RIDONDANZA 1

**Presenza di ridondanza**

**Spazio:** 4 byte \* 16.000.000 = 64.000.000.000 byte (4 byte perché memorizziamo un integer)

**Tempo:** 40 accessi/5 min

**Assenza di ridondanza**

**Spazio:** 0

**Tempo:** 40 accessi/5 min

**Decisione:** Siccome i tempi di accesso sono uguali, il problema da affrontare è se eliminare la ridondanza e avere tempi di accesso maggiori nella ridondanza successiva o sovraccaricare lo spazio nel database e avere tempi di accesso minore dopo. Abbiamo deciso di non eliminare la ridondanza perché ci escono dei risultati più accurati e specifici; inoltre ci aiuterebbe a rifare questa query con più semplicità.

2.3.1.2. numeroMediaSpect Streamer(attributo ricavabile da associazione)



**DERIVAZIONE:** L'attributo è derivabile dalla media del totale delle medie delle visualizzazioni delle live per il numero di live effettuate

**OPERAZIONI COINVOLTE**

**Op10:** Sommo il totale delle medie degli spettatori per il numero di live effettuate

**PRESENZA DI RIDONDANZA**

**Op10:** Sommo il totale delle medie degli spettatori per il numero di live effettuate

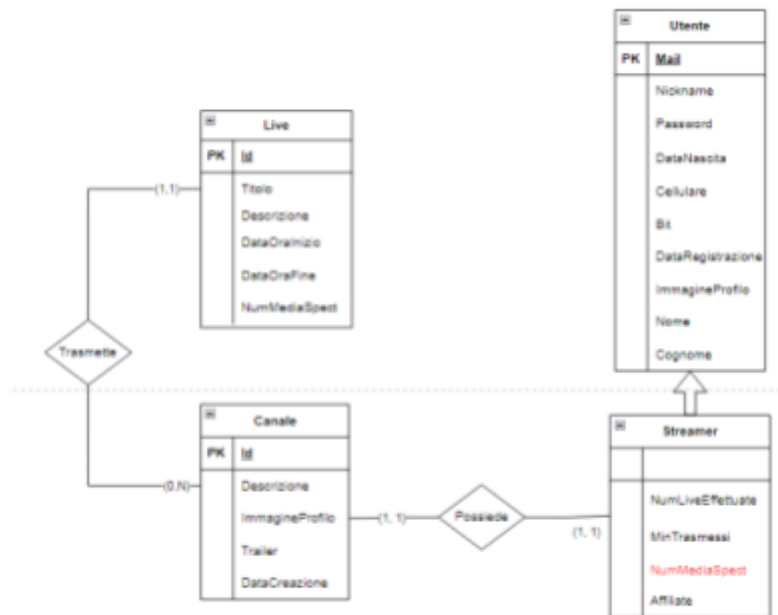


Tavola degli accessi:

| Concetto        | Costrutto    | Accessi | Tipo | Descrizione   |
|-----------------|--------------|---------|------|---|
| Live            | Entità       | 2       | L    | Primo accesso estraggo il numero medio di spettatori, al secondo accesso estraggo il numero di live fatte |
| Trasmette       | Associazione | 1       | L    | Solo per il passaggio   |
| Canale          | Entità       | 1       | L    | Eseguo una join per collegare l'entità Live   |
| Possiede        | Associazione | 1       | L    | Solo per il passaggio   |
| Streamer/Utente | Entità       | 1       | S    | Scrivo il numero medio di spettatori  |

S: 3 (entità) \* 1 (S) \* 1 (frequenza) = 3 accessi/giorno

L: 2 (associazione) \* 4 (L) \* 1 (frequenza) = 8 accessi/giorno

TOT: 3 + 8 = 11 accessi/giorno

### ASSENZA DI RIDONDANZA

**Op10:** Sommo il totale delle medie degli spettatori per il numero di live effettuate



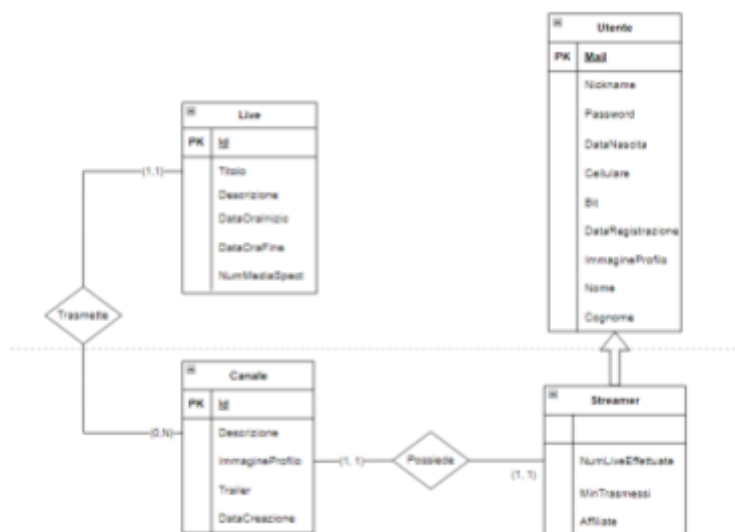


Tavola degli accessi:

| Concetto        | Costrutto    | Accessi | Tipo | Descrizione   |
|-----------------|--------------|---------|------|---|
| Live            | Entità       | 2       | L    | Primo accesso estraggo il numero medio di spettatori, al secondo accesso estraggo il numero di live fatte |
| Trasmette       | Associazione | 1       | L    | Solo per il passaggio   |
| Canale          | Entità       | 1       | L    | Eseguo una join per collegare l'entità Live   |
| Possiede        | Associazione | 1       | L    | Solo per il passaggio   |
| Streamer/Utente | Entità       | 1       | L    | Mostro a video il risultato della query   |

S: 3 (entità) \* 0 (S) \* 0 (frequenza) = 0 accessi/giorno

L: 2 (associazione) \* 5 (L) \* 1 (frequenza) = 10 accessi/giorno

TOT: 0 + 10 = 10 accessi/giorno

## TOTALI PER RIDONDANZA 2

### Presenza di ridondanza

**Spazio:** 4 \* 8.000.000 = 32.000.000 byte (4 byte perché memorizziamo un integer)

**Tempo:** 11 accessi/giorno

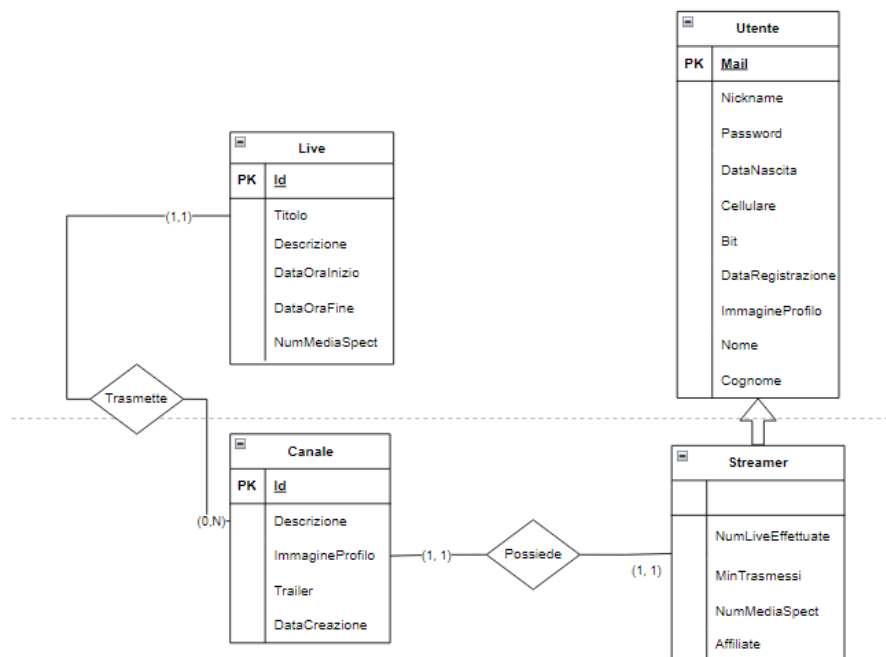
### Assenza di ridondanza

**Spazio:** 0

**Tempo:** 10 accessi/giorno

**Decisione:** Abbiamo deciso di eliminare la ridondanza perché ci sono meno tempi d'accesso e risparmiamo memoria sul database.

2.3.1.3. numeroLiveEffettuate Streamer(dal conteggio delle partecipazioni ad una associazione)



**DERIVAZIONE:** L'attributo è ricavabile dalla conta delle live effettuate sul canale dello Streamer

### OPERAZIONI COINVOLTE

**Op11:** Calcolare il numero delle live effettuate dallo streamer

### PRESENZA DI RIDONDANZA

**Op11:** Calcolare il numero delle live effettuate dallo streamer

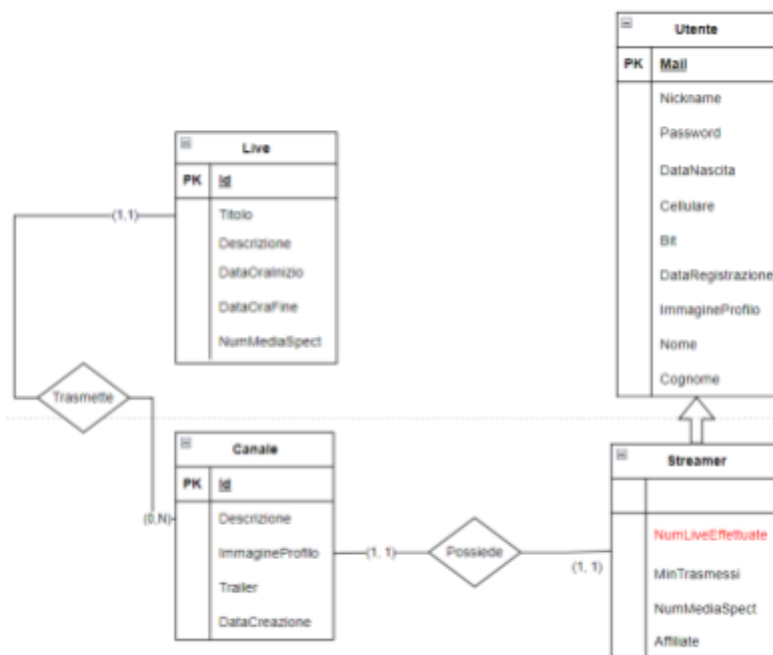


Tavola degli accessi:

| Concetto        | Costrutto    | Accessi | Tipo | Descrizione   |
|-----------------|--------------|---------|------|---|
| Live            | Entità       | 2       | L    | Primo accesso estraggo il numero medio di spettatori, al secondo accesso estraggo il numero di live fatte |
| Canale          | Entità       | 1       | L    | Eseguo una join per collegare l'entità Live   |
| Possiede        | Associazione | 1       | L    | Join per collegare l'utente al numero di live effettuate  |
| Streamer/Utente | Entità       | 1       | S    | Scrivo il numero di live effettuate   |

S: 3 (entità) \* 1 (S) \* 1 (frequenza) = 3 accessi/giorno

L: 1 (associazione) \* 3 (L) \* 1 (frequenza) = 3 accessi/giorno

TOT: 3 + 3 = 6 accessi/giorno

## ASSENZA DI RIDONDANZA

**Op11**: Calcolare il numero delle live effettuate dallo streamer

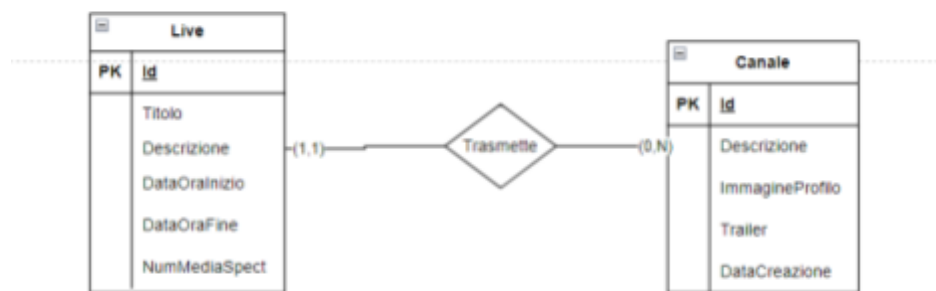


Tavola degli accessi:

| Concetto | Costrutto | Accessi | Tipo | Descrizione   |
|----------|-----------|---------|------|---|
| Live     | Entità    | 1       | L    | Estraggo la live che è iniziata e terminata   |
| Canale   | Entità    | 1       | L    | Eseguo una join per collegare l'entità Live ed ottenere il numero di live con una count |

S: 0 accessi/giorno

L:  $2 (L) * 1$  (frequenza) = 2 accessi/giorno

TOT: 2 accessi/giorno

### TOTALI PER RIDONDANZA 3

#### Presenza di ridondanza

**Spazio:**  $4 \text{ byte} * 8.000.000 = 32.000.000 \text{ byte}$  (4 byte perché memorizziamo un integer)

**Tempo:** 6 accessi/giorno

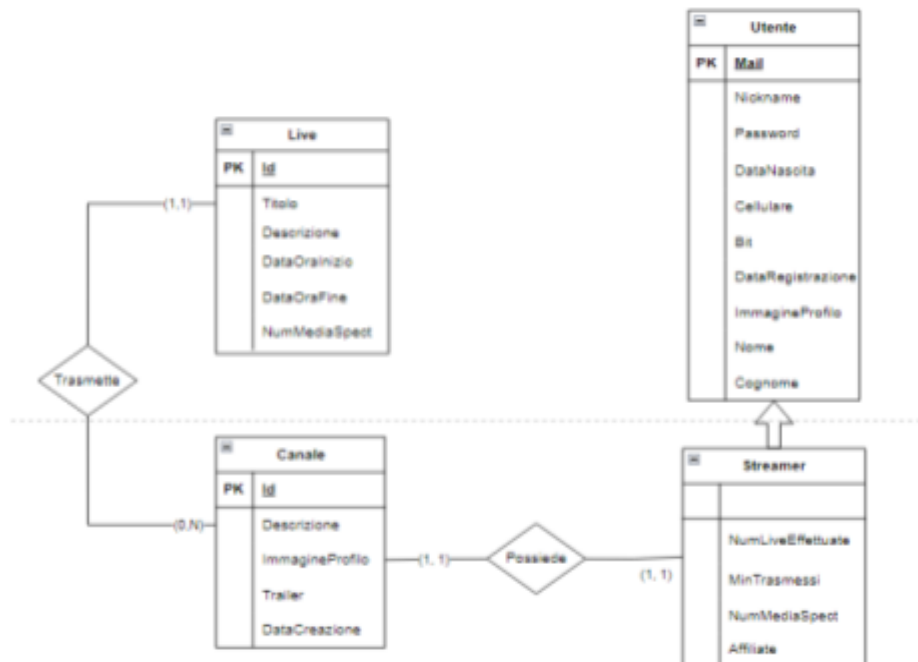
#### Assenza di ridondanza

**Spazio:** 0

**Tempo:** 2 accessi/giorno

**Decisione:** non ci sono dubbi sulla scelta di optare la soluzione senza ridondanza; il numero di accessi è quasi invariato (e, a vantaggio dell'assenza della ridondanza); il numero di byte risparmiati è notevole, soprattutto per una piattaforma di queste dimensioni.

#### 2.3.1.4. minTrasmesi Streamer (dal conteggio delle partecipazioni ad una associazione)



**DERIVAZIONE:** L'attributo è ricavabile dalla somma dei minuti totali delle varie live (ottenuti dalla differenza tra l'orario di fine e quello di inizio della live) effettuate sul canale dello streamer

#### OPERAZIONI COINVOLTE

**Op12:** Calcolare il numero di minuti totali delle live effettuate dallo streamer

#### PRESENZA DI RIDONDANZA

**Op12:** Calcolare il numero di minuti totali delle live effettuate dallo streamer

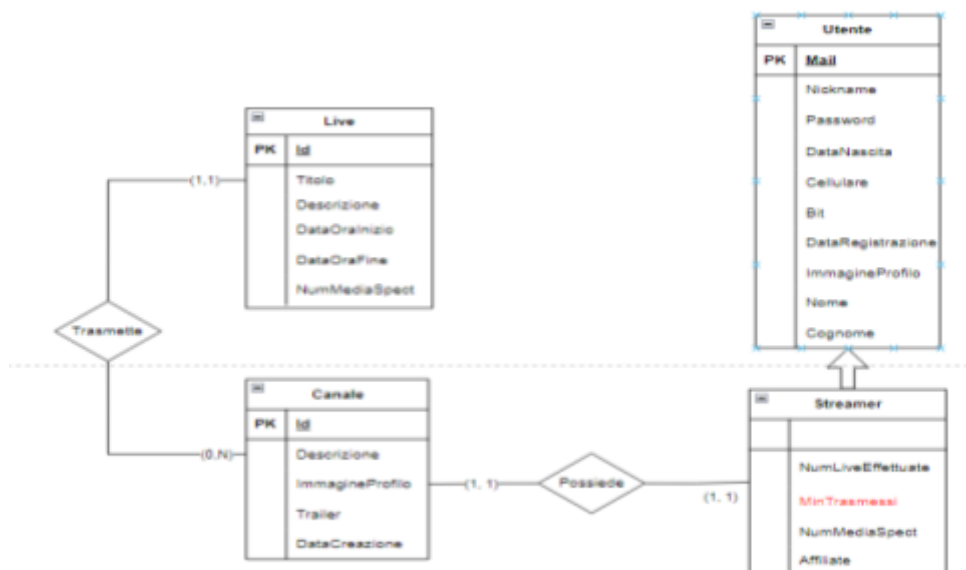


Tavola degli accessi:

| Concetto        | Costrutto    | Accessi | Tipo | Descrizione   |
|-----------------|--------------|---------|------|---|
| Live            | Entità       | 1       | L    | Estraggo il numero dei minuti trasmessi facendo la differenza tra DataOraFine e DataOraInizio |
| Canale          | Entità       | 1       | L    | Eseguo una join per collegare l'entità Live   |
| Possiede        | Associazione | 1       | L    | Join per collegare l'utente al numero dei minuti trasmessi in live                            |
| Streamer/Utente | Entità       | 1       | S    | Scrivo il numero dei minuti trasmessi in live   |

S:  $3 \text{ (entità)} * 1 \text{ (S)} * 1 \text{ (frequenza)} = 3 \text{ accessi/giorno}$

L:  $1 \text{ (associazione)} * 3 \text{ (L)} * 1 \text{ (frequenza)} = 3 \text{ accessi/giorno}$

TOT:  $3 + 3 = 6 \text{ accessi/giorno}$

## ASSENZA DI RIDONDANZA

**Op12:** Calcolare il numero di minuti totali delle live effettuate dallo streamer

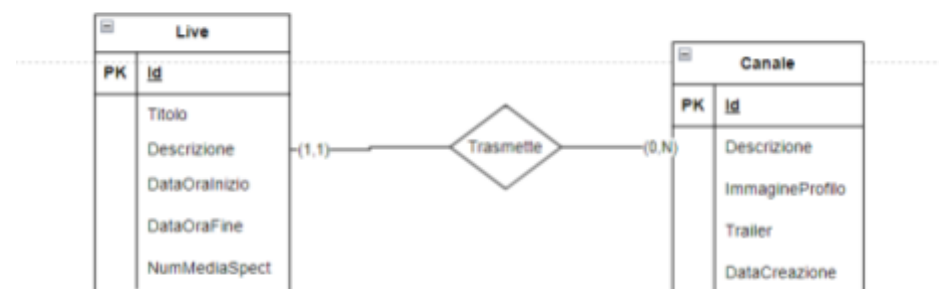


Tavola degli accessi:

| Concetto | Costrutto | Accessi | Tipo | Descrizione  |
|----------|-----------|---------|------|--|
| Live     | Entità    | 1       | L    | Estraggo il numero dei minuti trasmessi facendo la differenza tra DataOraFine e DataOraInizio                |
| Canale   | Entità    | 1       | L    | Eseguo una join per collegare l'entità Live ed ottenere il numero dei minuti trasmessi in live con una count |

S: 0 accessi/giorno

L:  $2 \text{ (L)} * 1 \text{ (frequenza)} = 2 \text{ accessi/giorno}$

TOT: 2 accessi/giorno

## TOTALI PER RIDONDANZA 4

### Presenza di ridondanza

**Spazio:** 4 byte \* 8.000.000 = 32.000.000 byte (4 byte perché memorizziamo un integer)

**Tempo:** 6 accessi/giorno

### Assenza di ridondanza

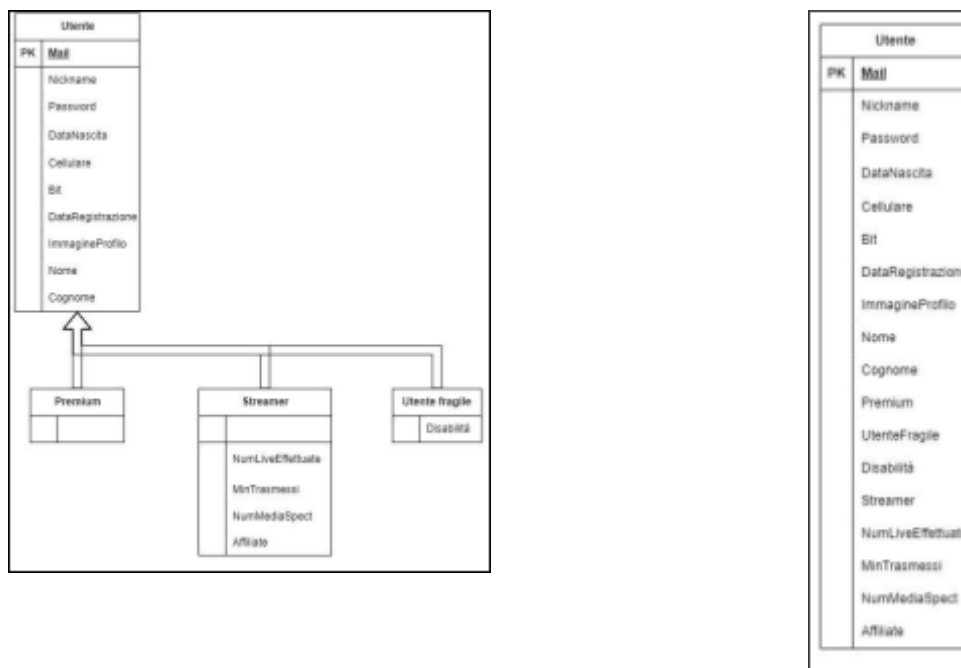
**Spazio:** 0

**Tempo:** 2 accessi/giorno

**Decisione:** non ci sono dubbi sulla scelta di optare la soluzione senza ridondanza; il numero di accessi è quasi invariato (e, a vantaggio dell'assenza della ridondanza); il numero di byte risparmiati è notevole, soprattutto per una piattaforma di queste dimensioni.

## 2.3.2. Eliminazione delle generalizzazioni

### 2.3.2.1. Generalizzazione 1 (<Utente>)



<Abbiamo accorpato le gerarchie di utente nel padre, così da avere tutte le informazioni dell'utente in un'unica entità; i campi streamer, premium e utente fragile sono campi booleani che indicano se l'utente copre uno di questi ruoli, mentre i campi dei figli accorpati al padre possono assumere il valore di NULL quando la loro specializzazione è a false>

## Regole aziendali

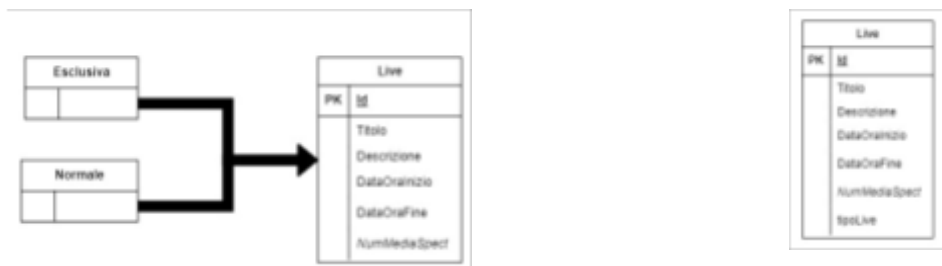
### Vincoli di Integrità:

|     |   |
|-----|---|
| RV1 | Utente può essere premium                     |
| RV2 | Utente può essere streamer                    |
| RV3 | Utente può essere fragile                     |
| RV4 | Utente può essere premium e streamer          |
| RV5 | Utente può essere premium e fragile           |
| RV6 | Utente può essere streamer e fragile          |
| RV7 | Utente può essere streamer, premium e fragile |

### Derivazioni:

|     |   |
|-----|---|
| RD1 | L'affiliazione di un utente si ottiene verificando i requisiti: numMediaSpect > 3, followers > 50 e numMinTrasmessi > 500 |
|-----|---|

#### 2.3.2.2. Generalizzazione 2 (<Live>)



<Abbiamo accorpato le gerarchie di Live dentro il padre, tramite l'utilizzo di un campo tipoLive, il quale può assumere solo due valori: 'esclusivo' se si tratta di una live esclusiva e 'normale' nel caso si tratti di una live normale >

## Regole aziendali

### Vincoli di Integrità:

|     |   |
|-----|---|
| RV1 | Live esclusiva deve essere vista solo da utenti premium |
| RV2 | Live deve essere relativa ad un canale                  |
| RV3 | Live deve essere pianificata da uno streamer            |
| RV4 | Live può essere esclusiva o normale                     |

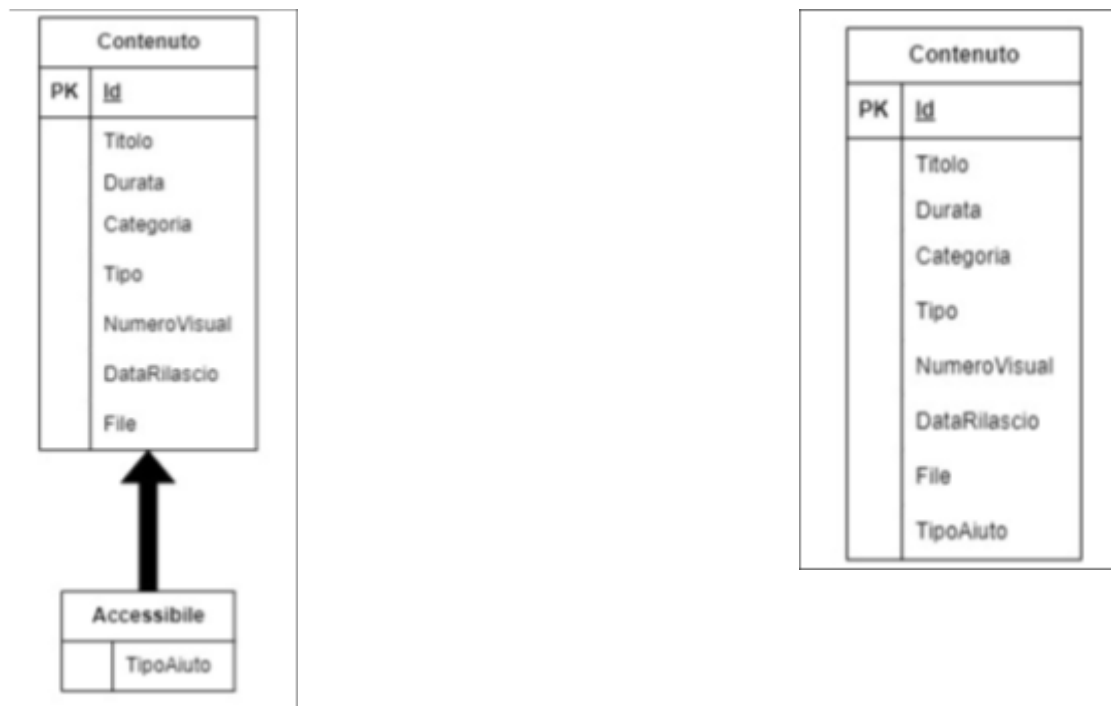
### Derivazioni:

|     |  |
|-----|--|
| RD1 | Il numero medio di spettatori si ottiene facendo la media di spettatori in live                          |
| RD2 | Il numero di live trasmesse da utente si ottiene facendo la somma di tutte le live trasmesse dall'utente |



|     |  |
|-----|--|
| RD3 | Il numero dei minuti trasmessi da utente si ottiene facendo la somma di tutti i minuti trasmessi in live dall'utente               |
| RD4 | Il numero di spettatori medi presenti si ottiene facendo la media del numero di persone che partecipano in live ogni minuto (live) |
| RD5 | Il numero di spettatori medi presenti si ottiene facendo la media del numero di persone che partecipano in tutte le live (utente)  |

### 2.3.2.3. Generalizzazione 3 (<Contenuto>)



<Abbiamo accorpato le gerarchie di Contenuto dentro il padre, e tramite il campo ‘tipoAiuto’ andiamo a verificare se si tratta di un contenuto accessibile o meno, infatti nel caso non sia accessibile il campo sarà impostato a NULL, mentre se ha un qualsiasi tipo di aiuto è accessibile>

## Regole aziendali

### Vincoli di Integrità:

|     |   |
|-----|---|
| RV1 | Contenuto deve essere relativo ad un canale               |
| RV2 | Contenuto può essere accessibile in base al tipo di aiuto |
| RV3 | Contenuto può essere video                                |
| RV4 | Contenuto può essere clip                                 |

### Derivazioni:

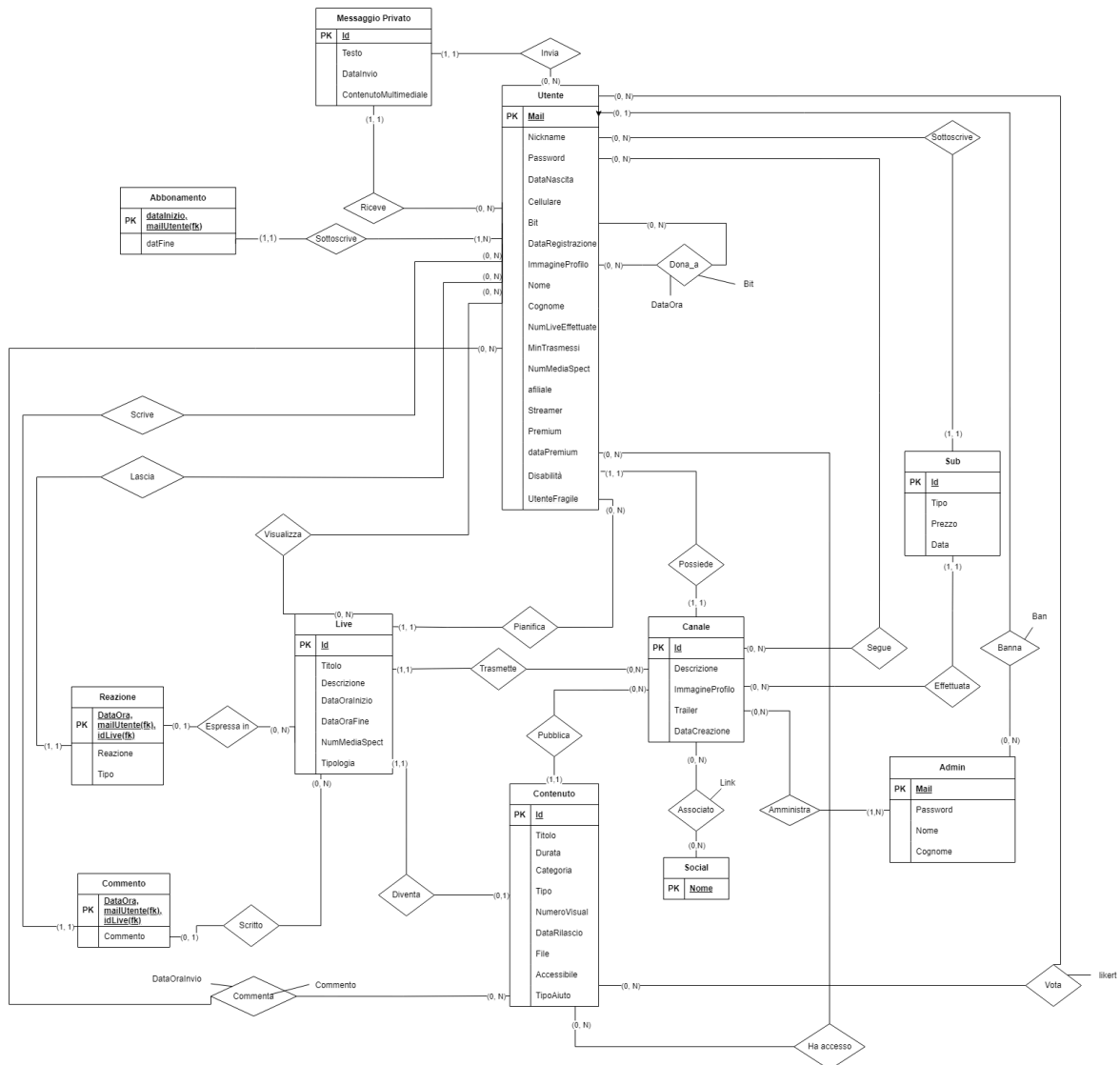
|     |   |
|-----|---|
| RD1 | Il numero di visualizzazioni di un contenuto si ottiene facendo la somma di tutti gli utenti che hanno visto quel contenuto |
|-----|---|

#### 2.3.4.scelta degli identificatori principali

| Entità           | Identificatore principale   |
|------------------|-----------------------------|
| Utente           | mail                        |
| Canale           | id                          |
| Live             | id                          |
| Contenuto        | id                          |
| Social           | nome                        |
| Admin            | utente                      |
| Reazione         | dataOra, mailUtente, idLive |
| Commento         | dataOra, mailUtente, idLive |
| MessaggioPrivato | id                          |
| Sub              | id                          |
| Abbonamento      | dataInizio, mailUtente      |

Abbiamo deciso di mettere un surrogato per Sub e Commento perché non potevamo usare una chiave primaria simile a reazione e commento siccome sul nostro e-r, gli attributi sono chiavi esterne. Su messaggio abbiamo usato una chiave surrogata perché la nostra idea era quella di mantenere la privacy tra gli utenti

## 2.4 Schema E-R ristrutturato + regole aziendali



### Regole aziendali

### Vincoli di Integrità:

|     |  |
|-----|--|
| RV1 | Messaggio Privato deve avere un solo mittente e un solo destinatario |
| RV2 | Live deve essere relativa ad un canale                               |
| RV3 | Live deve essere pianificata da uno streamer                         |
| RV4 | Sub deve essere sottoscritta da un utente                            |
| RV5 | Sub deve essere effettuata presso un canale                          |
| RV6 | Contenuto deve essere relativo ad un canale                          |

### Derivazioni:

|     |   |
|-----|---|
| RD1 | Il numero medio di spettatori si ottiene facendo la media di spettatori in live |
|-----|---|

|     |  |
|-----|--|
| RD2 | Il numero di live trasmesse da utente si ottiene facendo la somma di tutte le live trasmesse dall'utente                           |
| RD3 | Il numero dei minuti trasmessi da utente si ottiene facendo la somma di tutti i minuti trasmessi in live dall'utente               |
| RD4 | Il numero di spettatori medi presenti si ottiene facendo la media del numero di persone che partecipano in live ogni minuto (live) |
| RD5 | Il numero di spettatori medi presenti si ottiene facendo la media del numero di persone che partecipano in tutte le live (utente)  |
| RD6 | L'affiliate di un utente si ottiene verificando i requisiti: numMediaSpect > 3, followers > 50 e numMinTrasmessi > 500             |
| RD7 | Il numero di visualizzazioni di un contenuto si ottiene facendo la somma di tutti gli utenti che hanno visto quel contenuto        |

## 2.5 Schema relazionale con vincoli di integrità referenziale

**Utente**(mail, nome, cognome, immagineProfilo, nickname, password, dataNascita, Cellulare, Bit, DataRegistrazione, premium, streamer, numLive, minTrasmessi, NumMediaSpect, Affiliate, utenteFragile, disabilità)

**Abbonamento**(dataInizio, mailUtente, dataFine)

*Abbonamento(mailUtente) referencia Utente(mailUtente)*

**Canale**(id, descrizione, immagineProfilo, trailer, dataCreazione, ban, mailStreamer)

*Canale(mailStreamer) referencia Utente(mail)*

**Live**(id, titolo, descrizione, live, dataOraInizio, dataOraFine, numMediaSpect, tipologia, idCanale, mailStramer)

*Live(idCanale) referencia Canale(id)*

*Live(mailStramer) referencia Utente(mail)*

**Contenuto**(id, titolo, durata, categorie, tipo, numeroVisual, dataRilascio, likert, accessibile, idCanale)

*Contenuto(idCanale) referencia Canale(id)*

**Social**(nome)

**Admin**(utente, nome, cognome, password)

**Reazione**(dataOra, mailUtente, idLive, reazione, tipo)

*Reazione(mailUtente) referencia Utente(mail)*

*Reazione(idLive) referencia Live(id)*

**Commento**(dataOra, mailUtente, idLive, commento)

*Commento(mailUtente) referencia Utente(mail)*

*Commento(idLive) referencia Live(id)*

**MessaggioPrivato**(id, testo, dataInvio, contenutoMultimediale, *mailMittente*, *mailDestinatario*)

*MessaggioPrivato(mailMittente) referencia Utente(mail)*

*MessaggioPrivato(mailDestinatario) referencia Utente(mail)*

**Sub**(Id, tipo, prezzo, data, *mailUtente*, *idCanale*)

*Sub(mailUtente) referencia Utente(mail)*

*Sub(idCanale) referencia Canale(id)*

**visualizza**(*mailUtente*, *idLive*)

*Visualizza(mailUtente) referencia Utente(mail)*

*Visualizza(idLive) referencia Live(id)*

**dona\_a**(*mailUtente*, *mailStreamer*, bit)

*Dona\_a(mailUtente) referencia Utente(mail)*

*Dona\_a(mailStreamer) referencia Utente(mail)*

**segue**(*mailUtente*, *idCanale*)

*Segue(mailUtente) referencia Utente(mail)*

*Segue(idCanale) referencia Canale(id)*

**visualizzaEsclusiva**(*mailUtente*, *idLive*)

*visualizzaEsclusiva(mailUtente) referencia Utente(mail)*

*visualizzaEsclusiva(idLive) referencia Live(id)*

**vota**(*mailUtente*, *idContenuto*)

*Vota(mailUtente) referencia Utente(mail)*

*Vota(idContenuto) referencia Contenuto(id)*

**ha\_accesso**(*idContenutoAccessibile*, *mailUtenteFragile*)

*ha\_accesso(idContenutoAccessibile) referencia Contenuto(id)*

*ha\_accesso(mailUtenteFragile) referencia Utente(mail)*

**amministra**(*userAdmin*, *idCanale*)

*amministra(userAdmin) referencia Admin(utente)*

*amministra(idCanale) referencia Canale(id)*

**associato**(*idCanale*, *nomeSocial*, link)

*associato(idCanale) referencia Canale(id)*

*associato(nomeSocial) referencia social(nome)*

**commenta**(*mailUtente*, *idContenuto*, *dataOraInvio*, commento)

*commenta(mailUtente) referencia Utente(mail)*

*commenta(idContenuto) referencia Contenuto(id)*

## 3 DDL E DML

### 3 DDL di creazione del database

È stata adottata un approccio di progettazione del database con tabelle interconnesse mediante vincoli relazionali, utilizzando chiavi primarie e chiavi esterne per garantire l'integrità dei dati. L'identificazione univoca degli elementi è stata gestita tramite l'attributo "SERIAL" per gli identificatori delle chiavi primarie e i codici delle opere, mentre l'attributo "NOT NULL" è stato applicato alla colonna dell'indirizzo email. Siccome i nickname degli utenti devono essere diversi, abbiamo usato l'attributo "UNIQUE".

Di seguito un esempio generico di come sono state create le tabelle:

```
CREATE nomeTabella(  
    mail text PRIMARY KEY,  
    nickname text,  
    nome text,  
    cognome text,  
);
```

### 4 DML di popolamento di tutte le tabelle del database

Per il popolamento delle tabelle nel nostro database, abbiamo utilizzato il comando SQL INSERT INTO. Abbiamo dedicato particolare attenzione a rendere i dati il più verosimili possibile per riflettere scenari realistici. Ad esempio:

```
INSERT INTO nomeTabella (campo1, campo2, ..., campoN)  
  
VALUES  
  
('valore1', 'valore2', ..., 'valoreN');
```

Non tutti i campi sono obbligatori (NOT NULL), e ciò ci ha permesso di gestire in modo flessibile dati che potrebbero non essere disponibili in tutte le situazioni. Abbiamo utilizzato il tipo di dato VARCHAR per le colonne che contengono link a file, soprattutto se questi link sono gestiti da terze parti e la loro lunghezza potrebbe variare.

Per registrare informazioni temporali legate a determinate operazioni, come la creazione di account, la pubblicazione di contenuti o l'invio di messaggi, abbiamo utilizzato la funzione NOW()::DATE per inserire la data corrente in modo preciso.

Adottando queste pratiche, abbiamo mirato a garantire la coerenza e la veridicità dei dati nel nostro database, facilitando nel contempo operazioni di test e simulazioni che rispecchiano situazioni reali.

## **5 DML di modifica**

Abbiamo eseguito le operazioni di UPDATE delle tabelle per, ad esempio, per la verifica dell'affiliate (se rispetta determinati parametri), depositare dei bit sul portafoglio di un utente. Esiste anche l'operazione di DELETE per cancellare determinate informazioni. Abbiamo deciso di mettere ON DELETE CASCADE per eliminare le chiavi esterne.

Per modificare o cancellare abbiamo usato il secondo schema:

- UPDATE nomeTabella  
SET colonna1 = valore1, colonna2 = valore2, ...  
WHERE condizione;
- DELETE FROM nomeTabella WHERE condizione;