

Progetto di Basi di dati

Luca Barra, Jasmine Bizzo

Anno accademico 2022/2023

Indice

Progettazione concettuale	2
1.1 Requisiti iniziali	2
1.2 Glossario dei termini	4
1.3 Requisiti rivisti	4
1.4 Gruppi di frasi omogenee	6
1.5 Schema E-R principale + <i>business rules</i>	7
Progettazione logica	9
2.1 Tavola dei volumi	9
2.2 Tavola delle operazioni	11
2.3 Ristrutturazione dello schema E-R	12
2.3.1 Analisi delle ridondanze	12
2.3.2 Eliminazione delle generalizzazioni	14
2.3.3 Eliminazione degli attributi composti e multivalore	14
2.4 Schema E-R ristrutturato + <i>business rules</i>	15
2.5 Schema relazionale + vincoli di integrità referenziale	17
Implementazione	19
3.1 DDL di creazione del database	19
3.2 DML di popolamento di tutte le tabelle del database	24
3.3 Operazioni	26

Progettazione concettuale

1.1 Requisiti iniziali

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, 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 (a esempio Instagram, YouTube, ecc.), un'immagine profilo e anche un trailer (Figura 1(a)). In ogni canale possono esserci live, video (live passate) e clip (video di durata breve). Le live possono anche non diventare video del canale. Ognuno ha un titolo, una durata, appartiene a una categoria (Figura 1(b)) e può essere associato a diversi tag. Per ogni live viene memorizzato il numero medio di spettatori mentre per i video e le clip il numero di visualizzazioni.

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 followers), 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.

¹Liberamente ispirato a Twitch

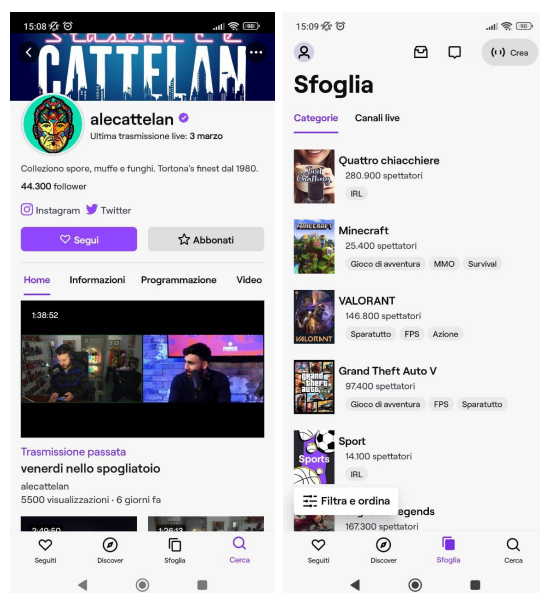


Figure 1: (a) La home del canale di alectattelan. (b) La lista delle categorie

Si può assumere che i contenuti multimediali vengano gestiti da una piattaforma di video hosting e che quindi sia sufficiente memorizzare un URL.

1.2 Glossario dei termini

Termine	Sinonimi	Breve descrizione	Termini collegati
Utente	Pubblico	Persona che usufruisce dei servizi della piattaforma	Spettatore registrato, Spettatore anonimo
Spettatore registrato	Viewer	Utente registrato che usufruisce dei contenuti di altri utenti	Utente, Streamer, Canale, Live, Bits
Spettatore anonimo	Viewer	Utente non registrato che usufruisce dei contenuti di altri utenti	Utente, Live
Streamer	Creatore di contenuti	Utente registrato che produce attivamente contenuti (live, video, clip)	Utente registrato, Canale, Live, Bits
Canale		Canale di uno streamer	Live, Video, Utente, Streamer
Live	Live Streaming, Dirette, Stream, Contenuti multimediali	Servizio video per interagire con il pubblico in tempo reale	Utente registrato, Utente anonimo, Streamer, Video, Clip
Video	Contenuti multimediali	Live passata	Canale, Live
Clips	Contenuti multimediali	Video di breve durata	Canale, Live
Bits		Valuta virtuale propria della piattaforma	Spettatore registrato, Streamer

1.3 Requisiti rivisti

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)~~ La **live** permette di interagire con ~~il pubblico~~ **gli utenti** in tempo reale grazie a feed video, chat e altro.

Ogni utente può essere spettatore o streamer, o entrambi. Gli spettatori possono essere **spettatori** registrati al servizio oppure ~~possono guardare le live in modo anonimo~~ essere spettatori anonimi. Ogni spettatore anonimo ha un **IP**. ~~Per registrarsi, gli utenti devono indicare~~ Ogni utente registrato ha un nome

utente, una password, una data di nascita, e un numero di telefono o un indirizzo mail. ~~Gli utenti iscritti possono~~ Ogni utente registrato può chattare, seguire lo streamer, creare ~~dirette~~ live.

~~Gli streamer hanno ciascuno~~ Ogni streamer ha un canale. ~~che può essere~~ caratterizzato tramite Ogni canale ha un nome e può avere una descrizione. ~~Per ogni canale, è possibile specificare~~ ha un numero di follower e può avere una lista di social associati (a esempio Instagram, YouTube, ecc.), un'immagine profilo e ~~anche~~ un trailer. ~~In~~ ogni canale ~~possono esserci~~ può avere live, video (live passate) e clip (video di durata breve). ~~Le~~ Una live ~~possono anche~~ può non diventare video del canale. ~~Ognuno~~ Ogni video ha un titolo, una durata, appartiene a una categoria e può essere associato a diversi tag. ~~Per ogni live viene memorizzato il~~ Ogni live ha un URL, un titolo, un numero medio di spettatori e degli orari. ~~mentre per i~~ Ogni video ha un numero di visualizzazioni e un URL. ~~le~~ Ogni clip il ha un numero di visualizzazioni e un URL.

~~Per ogni creatore di contenuti, si memorizzano il~~ Ogni streamer ha un numero di live effettuate, ~~il un~~ numero di minuti trasmessi, ~~un calendario il e un~~ numero medio di spettatori simultanei. ~~Inoltre, sulla pagina del canale viene visualizzato il numero di follower.~~

Quando uno streamer ~~rispetta determinati parametri di performance (~~ ha un minimo di 500 minuti trasmessi, una media di tre o più spettatori simultanei, e almeno 50 followers), 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.~~ In ogni calendario, uno streamer, può indicare orari e titolo delle live future.

~~I viewer possono~~ Ogni spettatore registrato può diventare follower del canale degli di uno o più streamer ~~che preferiscono, e le loro preferenze~~ Le preferenze degli utenti registrati sono raccolte in un elenco di followee a cui possono accedere dal loro profilo. ~~I viewer possono inoltre~~ Ogni spettatore registrato può supportare gli streamer tramite la subscription (a pagamento) al loro canale, ottenendo dei privilegi (emoticon personalizzate, ecc.). ~~Inoltre, gli utenti hanno~~ Ogni utente registrato ha 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.~~ Ogni utente registrato può messaggiare con altri utenti registrati.

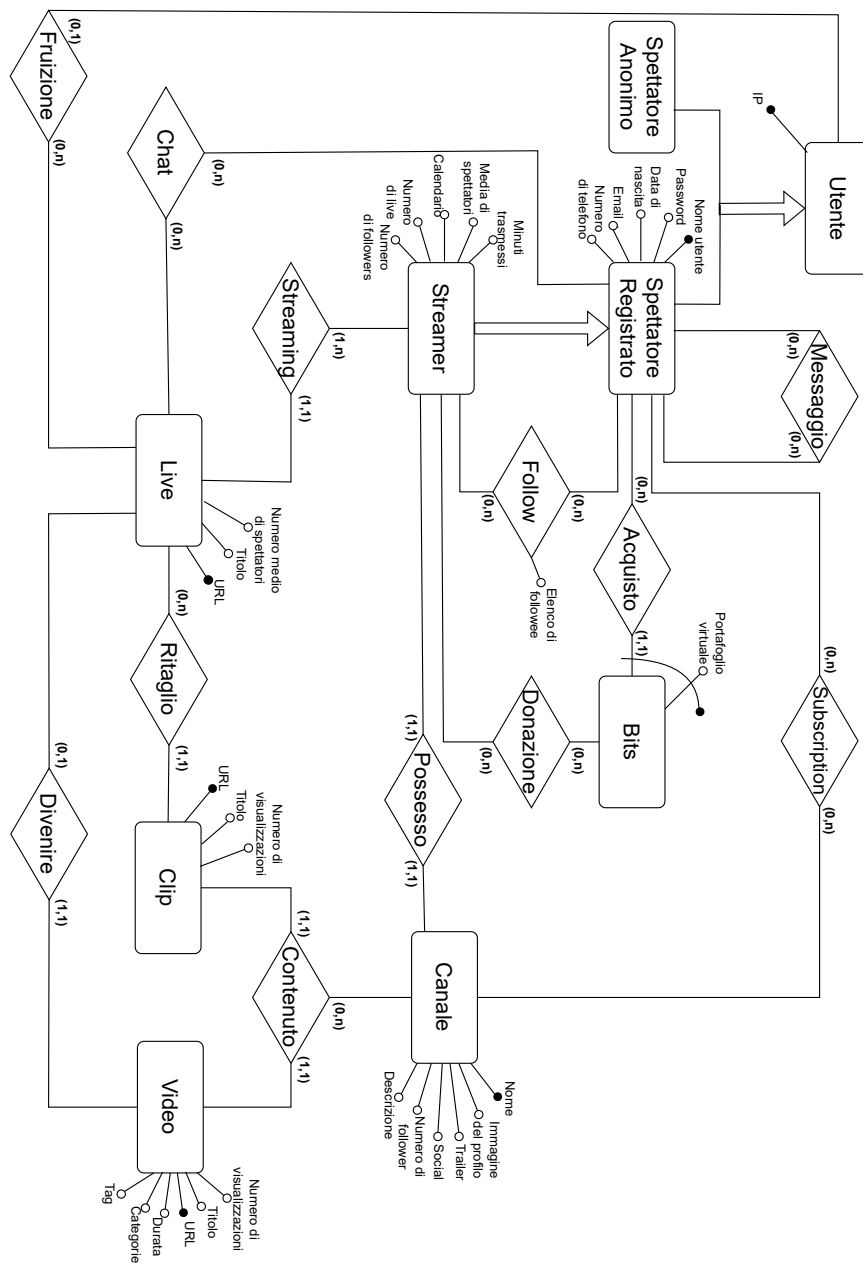
La base di dati deve supportare le seguenti operazioni:

- 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.~~

1.4 Gruppi di frasi omogenee

- Frasi di carattere generale: Si vuole realizzare una base di dati per un servizio che permette di fare live su vari argomenti;
- Frasi relative alle live: La live permette di interagire con gli utenti in tempo reale grazie a feed video, chat e altro. Una live può non diventare video del canale. Ogni live ha un URL, un titolo, un numero medio di spettatori e degli orari;
- Frasi relative agli utenti: Ogni utente può essere spettatore o streamer, o entrambi. Ogni utente registrato ha un nome utente, una password, una data di nascita, e un numero di telefono o un indirizzo mail. Ogni utente registrato può chattare, seguire lo streamer, creare live. Le preferenze degli utenti registrati sono raccolte in un elenco a cui possono accedere dal loro profilo. Ogni utente registrato ha un portafoglio di bit (moneta virtuale che possono acquistare tramite la piattaforma), che possono usare per fare donazioni agli streamer. Ogni utente registrato può messaggiare con altri utenti registrati;
- Frasi relative agli streamer: Ogni streamer ha un canale. Ogni streamer ha un numero di live effettuate, un numero di minuti trasmessi, un calendario e un numero medio di spettatori simultanei. Quando uno streamer ha un minimo di 500 minuti trasmessi, una media di tre o più spettatori simultanei e almeno 50 follower può diventare affiliate;
- Frasi relative agli spettatori registrati: Ogni spettatore registrato può diventare follower del canale di uno o più streamer. Ogni spettatore registrato può supportare gli streamer tramite la subscription (a pagamento) al loro canale, ottenendo dei privilegi (emoticon personalizzate, ecc.);
- Frasi relative agli spettatori anonimi: Ogni spettatore anonimo ha un IP;
- Frasi relative ai video: Ogni video ha un titolo, una durata, appartiene a una categoria e può essere associato a diversi tag. Ogni video ha un numero di visualizzazioni e un URL;
- Frasi relative alle clip: Ogni clip ha un numero di visualizzazioni e un URL;
- Frasi relative ai canali: Ogni canale ha un nome e può avere una descrizione. Ogni canale ha un numero di follower e può avere una lista di social associati (a esempio Instagram, YouTube, ecc.), un'immagine profilo e un trailer. Ogni canale può avere live, video e clip;
- Frasi relative ai calendari: In ogni calendario, uno streamer, può indicare orari e titolo delle live future.

1.5 Schema E-R principale + *business rules*

Business rules:

- Uno spettatore anonimo non può essere nè uno spettatore registrato nè uno streamer e non può creare live;
- Un utente registrato (spettatore registrato o streamer) deve avere almeno uno tra numero di telefono o email;
- Per completare l'iscrizione la data di nascita deve indicare un soggetto con età superiore o uguale a 14 anni;
- Uno specifico numero di telefono o uno specifico indirizzo email può essere associato a un solo utente registrato;
- Se uno spettatore ha pagato la subscription a uno streamer, lo spettatore ottiene dei privilegi (emoticon personalizzate, etc.);
- Quando un utente registrato lascia il follow a uno streamer il numero di followers di quello streamer e il numero di follow nella lista dell'utente registrato aumentano di uno. Quando un utente registrato fa un unfollow di uno streamer il numero di follower di quello streamer e il numero di follow nella lista dell'utente registrato diminuiscono di uno;
- Quando un utente fa una donazione o uno spettatore fa una subscription il valore di bit del suo portafoglio diminuisce e il valore di bit del portafoglio dello streamer aumenta della stessa quantità.

Documentazione aggiuntiva - cardinalità degli attributi:

- Utente: IP (id);
- Spettatore Registrato: Nome Utente (id), Password (1,1), Elenco di Followee (1,1), Data di Nascita (1,1), Numero di Telefono (0,1), Indirizzo Email (0,1);
- Streamer: Minuti Trasmessi (1,1), Media di Spettatori (1,1), Calendario (1,1), Numero di Live (1,1), Numero di Followers (1,1);
- Canale: Nome (id), Numero di Followers (1,1), Numero di Subs (1,1), Immagine del Profilo (0,1), Trailer (0,1), Descrizione (0,1), Social (0,n);
- Live: URL (id), Titolo (1,1), Numero Medio di Spettatori (1,1);
- Video: URL (id), Titolo (1,1), Numero di Visualizzazioni (1,1), Durata (1,1), Categoria (1,1), Tag (0,n);
- Clips: URL (id), Titolo (1,1), Numero di Visualizzazioni (1,1);
- Bits: Portafoglio Virtuale (1,1).

Progettazione logica

2.1 Tavola dei volumi

Concetto	Volume	Tipo	Motivi della scelta
Utente	44.000.000	E	Sia gli spettatori registrati sia quelli anonimi
Spettatore Registrato	40.000.000	E	Stima basata sul numero di utenti di Twitch a fine 2021
Spettatore Anonimo	4.000.000	E	Si stima una percentuale del 10% di spettatori anonimi
Streamer	7.000.000	E	Stima basata sul numero di utenti che fanno live su Twitch a fine 2021
Canale	7.000.000	E	Dato che a ogni streamer corrisponde un canale
Live	50.000.000	E	Partendo dal presupposto che tutti gli streamer facciano live
Video	45.000.000	E	Prevedendo che molte live possano divenire video
Clip	60.000.000	E	Supponendo che da ogni live vengano tratte circa due clip
Bits	500.000.000	E	Sapendo che non tutti li acquistano e che ne vendono minimo è 95

Follow	400.000.000	A	Supponendo che ogni spettatore segua circa 10 streamer
Subscription	80.000.000	A	Supponendo che ogni spettatore dia una subscription a due canali
Messaggio	100.000.000	A	Poichè non tutti messaggiano, ma alcuni spettatori scrivano di frequente
Acquisto	120.000.000	A	Non tutti lo fanno, ma acquistano bits frequentemente
Possesso	7.000.000	A	Poichè ogni streamer possiede un canale
Streaming	30.000.000	A	Poichè ogni live "è in diretta" una sola volta
Contenuto	700.000.000	A	Supponendo che ogni canale contenga in media 100 fra clip e video
Fruizione	440.000.000	A	Immaginando che gli utenti guardino in media 10 live a testa
Chat	100.000.000	A	Non tutti usano la chat, ma alcuni spettatori lo fanno di frequente
Divenire	45.000.000	A	Supponendo che la maggior parte delle live diventino video

2.2 Tavola delle operazioni

Operazione	Breve descrizione	Tipo	Frequenza
1	Si effettua la ricerca di una live	I	500.000/Giorno
2	Si effettua la ricerca di uno streamer	I	400.000/Giorno
3	Viene avviata una live	I	50.000/Giorno
4	Un utente paga la subscription a uno streamer	I	2.500/Giorno
5	Viene calcolato il numero medio di utenti attivi durante il mese	B	1/Mese
6	Viene calcolata la classifica dei canali con più sub	B	1/Settimana
7	Controlla le condizioni per la qualifica di affiliate	B	1/Giorno
8	Viene calcolata la classifica degli streamers più seguiti	B	1/Settimana

2.3 Ristrutturazione dello schema E-R

2.3.1 Analisi delle ridondanze

Ridondanza 1: operazione 6.

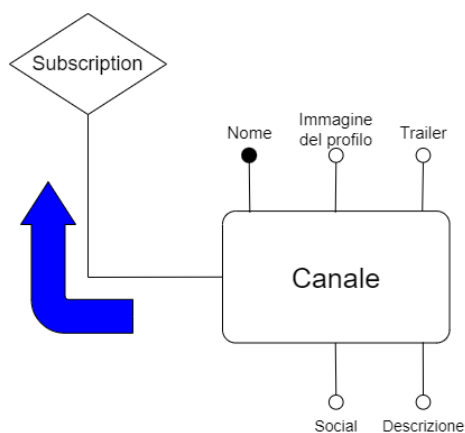
Concetto	Volume	Tipo
Canale	7.000.000	E
Subscription	80.000.000	A

Analisi senza ridondanza :

- $CARD(SUBSCRIPTION): 8 * 10^7$, $CARD(CANALE): 7 * 10^6$;
- Spazio: 0 byte.

Costrutto	Concetto	Tipo	Accessi
Subscription	Associazione	L	1
Canale	Entità	L	1

Si deve accedere in lettura n volte, dove n è il numero di streamers, quindi $CARD(SUBSCRIPTION) * CARD(CANALE) \approx 10^{14}$ accessi.

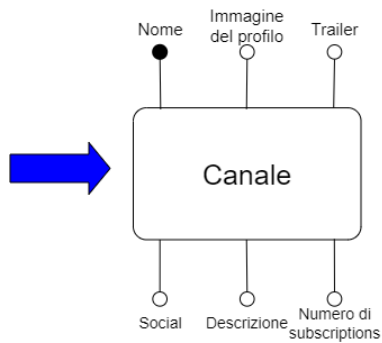


Analisi con ridondanza:

- $CARD(CANALE): 7 * 10^6$;
- Spazio: supponendo di usare 5 byte per l'attributo numero di subs si hanno circa 35 MBytes in più.

Costrutto	Concetto	Tipo	Accessi
Canale	Entità	L	1

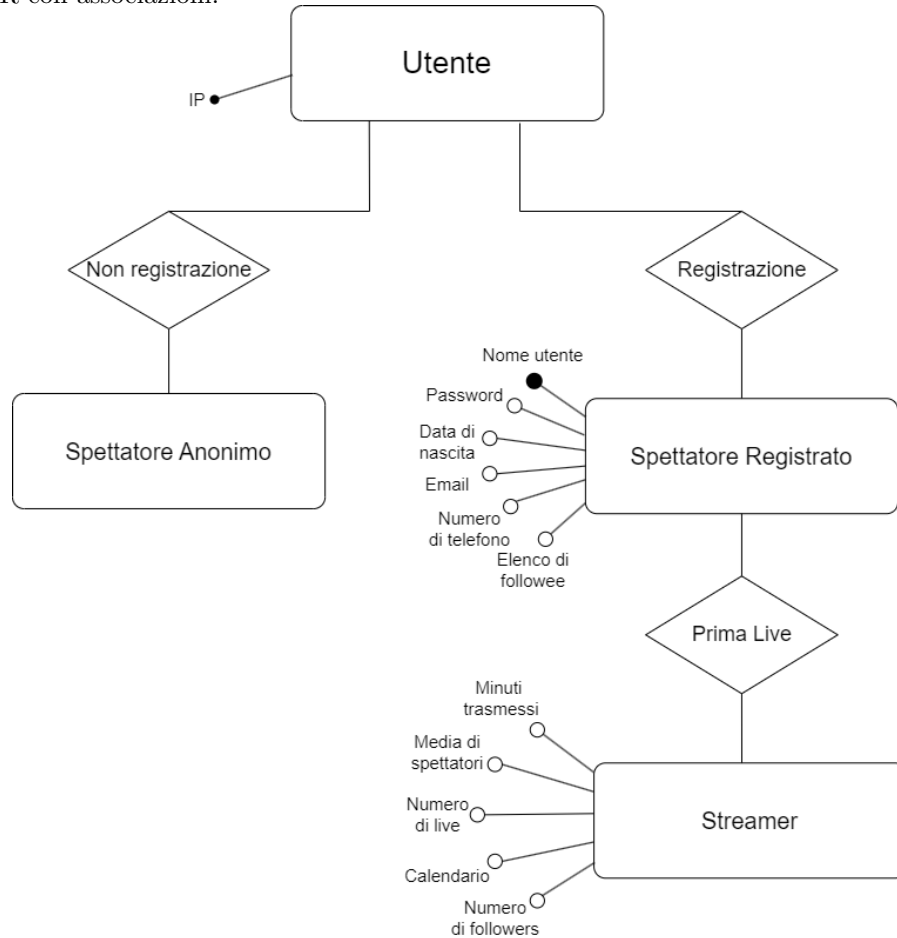
Si legge l'attributo numero di subs di ogni canale, quindi $CARD(CANALE) \approx 10^6$ accessi.



Conclusione: la ridondanza presenta un costo contenuto (appena 35 MBytes) a fronte di un evidente vantaggio in termini di accessi in lettura, quindi si è scelto di mantenerla.

2.3.2 Eliminazione delle generalizzazioni

Si è scelto di sostituire entrambe le generalizzazioni presenti nel primo schema E-R con associazioni.

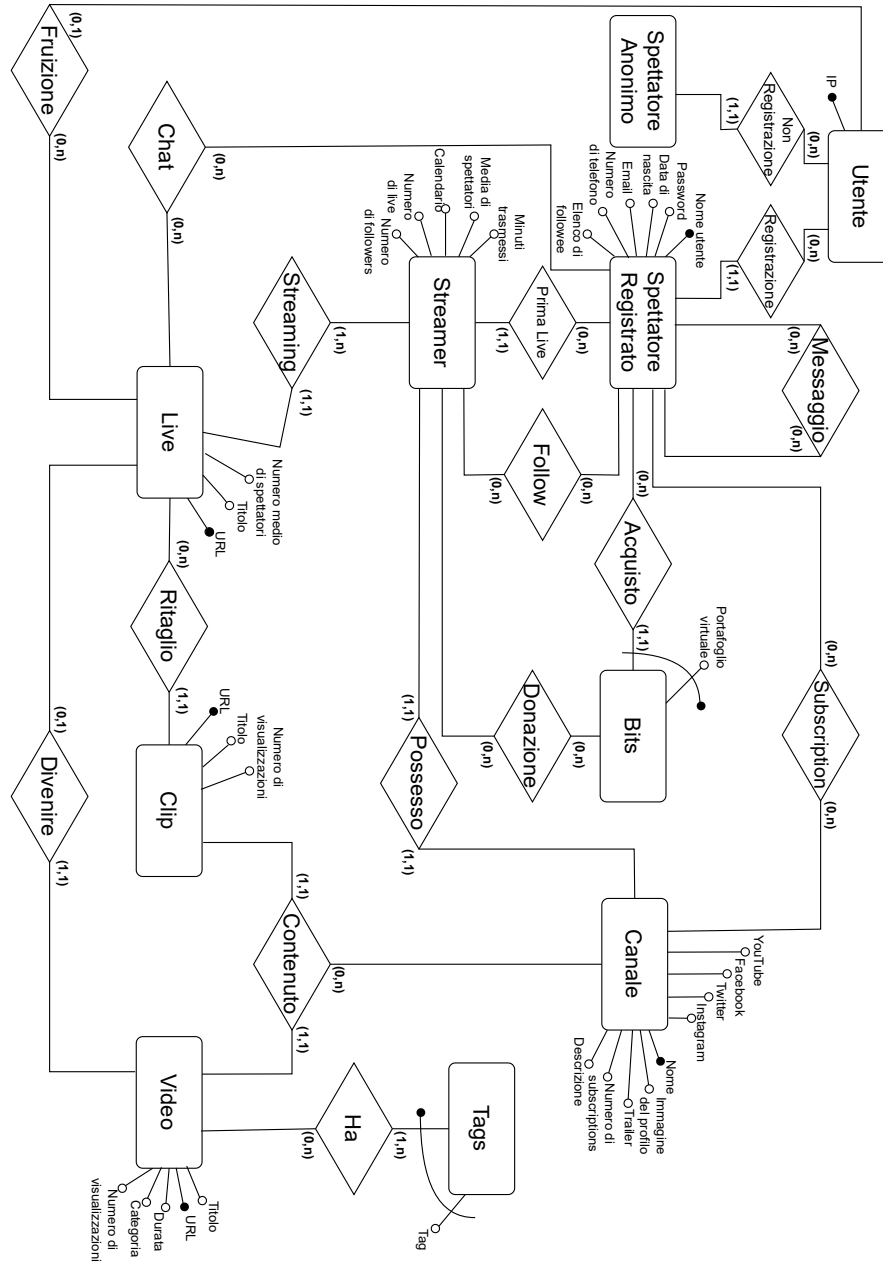


2.3.3 Eliminazione degli attributi composti e multivalore

Erano presenti due attributi multivalore (Social e Tag). Poichè lo schema relazionale non può gestire questo genere di attributi abbiamo scelto di percorrere due strade diverse.

Si è scelto di trattare "Social" come attributo composto, trasformandolo dunque in più attributi: Facebook, Twitter, Instagram e YouTube, ognuno dei quali può essere fornito tramite link o impostato come NULL.

Invece, l'attributo "Tag" è stato trasformato in un'entità Tags a cui viene assegnato un attributo Tag.

2.4 Schema E-R ristrutturato + *business rules*

Business rules:

- Un utente registrato (spettatore registrato o streamer) deve avere almeno uno tra numero di telefono o email;
- Per completare l'iscrizione la data di nascita deve indicare un soggetto con età superiore o uguale a 14 anni;
- Uno specifico numero di telefono o uno specifico indirizzo email può essere associato a un solo utente registrato;
- Se uno spettatore ha pagato la subscription a uno streamer, lo spettatore ottiene dei privilegi (emoticon personalizzate, etc.);
- Quando un utente registrato lascia il follow a uno streamer il numero di followers di quello streamer e il numero di follow nella lista dell'utente registrato aumentano di uno. Quando un utente registrato fa un unfollow di uno streamer il numero di follower di quello streamer e il numero di follow nella lista dell'utente registrato diminuiscono di uno;
- Quando un utente fa una donazione o uno spettatore fa una subscription il valore di bit del suo portafoglio diminuisce e il valore di bit del portafoglio dello streamer aumenta della stessa quantità.

Documentazione aggiuntiva - cardinalità degli attributi:

- Utente: IP (id);
- Spettatore Registrato: Nome Utente (id), Password (1,1), Elenco di Followee (1,1), Data di Nascita (1,1), Numero di Telefono (0,1), Indirizzo Email (0,1);
- Streamer: Minuti Trasmessi (1,1), Media di Spettatori (1,1), Calendario (1,1), Numero di Live (1,1), Numero di Followers (1,1);
- Canale: Nome(id), Numero di Followers(1,1), Numero di Subs (1,1), Immagine del Profilo (0,1), Trailer (0,1), Descrizione (0,1), *Facebook* (0,1), *Twitter* (0,1), *Instagram* (0,1), *YouTube* (0,1);
- Live: URL (id), Titolo (1,1), Numero Medio di Spettatori (1,1);
- Video: URL (id), Titolo (1,1), Numero di Visualizzazioni (1,1), Durata (1,1), Categoria (1,1);
- Clips: URL (id), Titolo (1,1), Numero di Visualizzazioni (1,1);
- Bits: Portafoglio Virtuale (1,1);
- *Tags*: *Tag* (1,1).

2.5 Schema relazionale + vincoli di integrità referenziale

Utente(IP);
 Spettatore Registrato(IP, Nome utente, Password, Data di nascita, Email, Numero di telefono, Elenco di followee);
 Spettatore Anonimo(IP);
 Streamer(IP, Nome utente, Minuti trasmessi, Media di spettatori, Calendario, Numero di live, Numero di followers);
 Canale(Nome, Immagine del profilo, Trailer, Social, Numero di subscriptions, Descrizione, Facebook, Twitter, Instagram, YouTube);
 Live(URL, Titolo, Numero medio di spettatori);
 Video(URL, Titolo, Numero di visualizzazioni_Video, Durata, Categorie, Tag);
 Clip(URL, Titolo, Numero di Visualizzazioni_Clip);
 Bits(Spettatore registrato, Portafoglio virtuale);
 Tags(Tag, Video);
 Registrazione(Utente, Spettatore registrato);
 Non-Registrazione(Utente, Spettatore Anonimo);
 Prima Live(Spettatore registrato, Streamer);
 Follow(Spettatore registrato, Streamer);
 Subscription(Spettatore registrato, Canale);
 Messaggio(Spettatore Registrato_1, Spettatore Registrato_2);
 Acquisto(Spettatore Registrato, Bits);
 Possesso(Streamer, Canale);
 Streaming(Streamer, Live);
 Contenuto(Canale, Video, Clip);
 Fruizione(Utente, Live);
 Chat(Spettatore Registrato, Live);
 Divenire(Live, Video);
 Ha(Video, Tags).

Vincoli:

- Registrazione(Utente), Non-Registrazione(Utente) e Fruizione(Utente) referenziano Utente(IP);
- Bits(Spettatore registrato), Registrazione(Spettatore registrato), Prima Live(Spettatore registrato), Follow(Spettatore registrato), Subscription(Spettatore registrato), Messaggio(Spettatore Registrato_1), Messaggio(Spettatore Registrato_2), Acquisto(Spettatore Registrato) e Chat(Spettatore Registrato) referenziano Spettatore Registrato(Nome utente);
- Non-Registrazione(Spettatore Anonimo) riferenzia Spettatore Anonimo(IP);
- Prima Live(Streamer), Follow(Streamer), Possesso(Streamer) e Streaming(Streamer) referenziano Streamer(Nome utente);

- Subscription(Canale), Possesso(Canale), Tags(Canale), Ha(Canale) e Contenuto(Canale) referenziano Canale(Nome);
- Acquisto(Bits) riferenzia Bits(Spettatore registrato);
- Streaming(Live), Fruizione(Live), Chat(Live) e Divenire(Live) referenziano Live(URL);
- Contenuto(Video), Ha(Video) e Divenire(Video) referenziano Video(URL);
- Contenuto(Clip) riferenzia Clip(URL);
- Ha(Tags) riferenzia Tags(Video).

Implementazione

3.1 DDL di creazione del database

UTENTE

```
CREATE TABLE Utente (  
    IP VARCHAR(15) PRIMARY KEY  
);
```

SPETTATORE REGISTRATO

```
CREATE TABLE SpettatoreRegistrato (  
    NomeUtente VARCHAR(50) PRIMARY KEY,  
    IP VARCHAR(15) UNIQUE,  
    Password VARCHAR(50) IS NOT NULL,  
    DataDiNascita DATE IS NOT NULL,  
    Email VARCHAR(100),  
    NumeroDiTelefono VARCHAR(100),  
    FOREIGN KEY (IP) REFERENCES Utente(IP)  
);
```

SPETTATORE ANONIMO

```
CREATE TABLE SpettatoreAnonimo (  
    IP VARCHAR(15) PRIMARY KEY,  
    FOREIGN KEY (IP) REFERENCES Utente(IP)  
);
```

STREAMER

```
CREATE TABLE Streamer (  
    NomeUtente VARCHAR(50),  
    IP VARCHAR(15) UNIQUE,  
    MinutiTrasmessi INTEGER,  
    MediaDiSpettatori INTEGER,  
    Calendario VARCHAR(200),  
    NumeroDiLive INTEGER,  
    NumeroDiFollowers INTEGER  
    FOREIGN KEY (IP) REFERENCES Utente(IP)  
);
```

CANALE

```
CREATE TABLE Canale (  
    Nome VARCHAR(100) PRIMARY KEY,  
    ImmagineDelProfilo VARCHAR(200),  
    Trailer VARCHAR(200),  
    Facebook VARCHAR(200),  
    Twitter VARCHAR(200),  
    Instagram VARCHAR(200),  
    YouTube VARCHAR(200),  
    NumeroDiSubscriptions INTEGER,
```

```
||     Descrizione VARCHAR(500)
|| );
```

LIVE

```
||     CREATE TABLE Live (
||     URL VARCHAR(200) PRIMARY KEY,
||     Titolo VARCHAR(100),
||     NumeroMedioDiSpettatori INTEGER
|| );
```

VIDEO

```
||     CREATE TABLE Video (
||     URL VARCHAR(200) PRIMARY KEY,
||     Titolo VARCHAR(100),
||     NumeroDiVisualizzazioniVideo INTEGER,
||     Durata TIME,
||     Categorie VARCHAR(200),
||     Tag VARCHAR(200)
|| );
```

CLIP

```
||     CREATE TABLE Clip (
||     URL VARCHAR(200) PRIMARY KEY,
||     Titolo VARCHAR(100),
||     NumeroDiVisualizzazioniClip INTEGER
|| );
```

BITS

```
||     CREATE TABLE Bits (
||     SpettatoreRegistrato VARCHAR(50),
||     PortafoglioVirtuale DECIMAL(10, 2),
||     FOREIGN KEY (SpettatoreRegistrato) REFERENCES
||         SpettatoreRegistrato(NomeUtente),
||     PRIMARY KEY (SpettatoreRegistrato, PortafoglioVirtuale)
|| );
```

TAGS

```
||     CREATE TABLE Tags (
||     Tag VARCHAR(200),
||     Video VARCHAR(200),
||     FOREIGN KEY (Video) REFERENCES Video(URL),
||     PRIMARY KEY (Tag, Video)
|| );
```

REGISTRAZIONE

```
||     CREATE TABLE Registrazione (
||     Utente VARCHAR(15),
||     SpettatoreRegistrato VARCHAR(50),
```

```
    FOREIGN KEY (Utente) REFERENCES Utente(IP),  
    FOREIGN KEY (SpettatoreRegistrato) REFERENCES  
        SpettatoreRegistrato(NomeUtente),  
    PRIMARY KEY (Utente, SpettatoreRegistrato)  
);
```

NON_REGISTRAZIONE

```
CREATE TABLE NonRegistrazione (  
    Utente VARCHAR(15),  
    SpettatoreAnonimo VARCHAR(15),  
    FOREIGN KEY (Utente) REFERENCES Utente(IP),  
    FOREIGN KEY (SpettatoreAnonimo) REFERENCES SpettatoreAnonimo(IP),  
    PRIMARY KEY (Utente)  
);
```

PRIMA LIVE

```
CREATE TABLE PrimaLive (  
    SpettatoreRegistrato VARCHAR(50),  
    Streamer VARCHAR(15),  
    FOREIGN KEY (SpettatoreRegistrato) REFERENCES  
        SpettatoreRegistrato(NomeUtente),  
    FOREIGN KEY (Streamer) REFERENCES Streamer(NomeUtente),  
    PRIMARY KEY (SpettatoreRegistrato, Streamer)  
);
```

FOLLOW

```
CREATE TABLE Follow (  
    SpettatoreRegistrato VARCHAR(50),  
    Streamer VARCHAR(15),  
    FOREIGN KEY (SpettatoreRegistrato) REFERENCES  
        SpettatoreRegistrato(NomeUtente),  
    FOREIGN KEY (Streamer) REFERENCES Streamer(NomeUtente),  
    PRIMARY KEY (SpettatoreRegistrato, Streamer)  
);
```

SUBSCRIPTION

```
CREATE TABLE Subscription (  
    SpettatoreRegistrato VARCHAR(50),  
    Canale VARCHAR(100),  
    FOREIGN KEY (SpettatoreRegistrato) REFERENCES  
        SpettatoreRegistrato(NomeUtente),  
    FOREIGN KEY (Canale) REFERENCES Canale(Nome),  
    PRIMARY KEY (SpettatoreRegistrato, Canale)  
);
```

MESSAGGIO

```
CREATE TABLE Messaggio (  
    SpettatoreRegistrato1 VARCHAR(50),
```

```
SpettatoreRegistrato2 VARCHAR(50),
FOREIGN KEY (SpettatoreRegistrato1) REFERENCES
    SpettatoreRegistrato(NomeUtente),
FOREIGN KEY (SpettatoreRegistrato2) REFERENCES
    SpettatoreRegistrato(NomeUtente),
PRIMARY KEY (SpettatoreRegistrato1, SpettatoreRegistrato2)
);
```

ACQUISTO

```
CREATE TABLE Acquisto (
    SpettatoreRegistrato VARCHAR(50),
    Bits DECIMAL(10, 2),
    FOREIGN KEY (SpettatoreRegistrato) REFERENCES
        SpettatoreRegistrato(NomeUtente),
    FOREIGN KEY (Bits) REFERENCES
        Bits(SpettatoreRegistrato)
    PRIMARY KEY (SpettatoreRegistrato, Bits)
);
```

POSSESSO

```
CREATE TABLE Possesso (
    Streamer VARCHAR(15) UNIQUE,
    Canale VARCHAR(100) UNIQUE,
    FOREIGN KEY (Streamer) REFERENCES Streamer(NomeUtente),
    FOREIGN KEY (Canale) REFERENCES Canale(Nome),
    PRIMARY KEY (Streamer, Canale)
);
```

STREAMING

```
CREATE TABLE Streaming (
    Streamer VARCHAR(15),
    Live VARCHAR(200) UNIQUE,
    FOREIGN KEY (Streamer) REFERENCES Streamer(NomeUtente),
    FOREIGN KEY (Live) REFERENCES Live(URL),
    PRIMARY KEY (Streamer, Live)
);
```

CONTENUTO

```
CREATE TABLE Contenuto (
    Canale VARCHAR(100),
    Video VARCHAR(200) UNIQUE,
    Clip VARCHAR(200) UNIQUE,
    FOREIGN KEY (Canale) REFERENCES Canale(Nome),
    FOREIGN KEY (Video) REFERENCES Video(URL),
    FOREIGN KEY (Clip) REFERENCES Clip(URL),
    PRIMARY KEY (Canale)
);
```

FRUIZIONE

```
CREATE TABLE Fruizione (  
  Utente VARCHAR(15),  
  Live VARCHAR(200),  
  FOREIGN KEY (Utente) REFERENCES Utente(IP),  
  FOREIGN KEY (Live) REFERENCES Live(URL),  
  PRIMARY KEY (Utente, Live)  
);
```

CHAT

```
CREATE TABLE Chat (  
  SpettatoreRegistrato VARCHAR(50),  
  Live VARCHAR(200),  
  FOREIGN KEY (SpettatoreRegistrato) REFERENCES  
    SpettatoreRegistrato(NomeUtente),  
  FOREIGN KEY (Live) REFERENCES Live(URL),  
  PRIMARY KEY (SpettatoreRegistrato, Live)  
);
```

DIVENIRE

```
CREATE TABLE Divenire (  
  Live VARCHAR(200) UNIQUE,  
  Video VARCHAR(200) UNIQUE,  
  FOREIGN KEY (Live) REFERENCES Live(URL),  
  FOREIGN KEY (Video) REFERENCES Video(URL),  
  PRIMARY KEY (Live, Video)  
);
```

HA

```
CREATE TABLE Ha (  
  Tags VARCHAR(200),  
  Video VARCHAR(200),  
  FOREIGN KEY (Tags) REFERENCES Tags(Tag),  
  FOREIGN KEY (Video) REFERENCES Video(URL),  
  PRIMARY KEY (Tags, Video)  
);
```

3.2 DML di popolamento di tutte le tabelle del database

Le tabelle che sono generate da associazioni non hanno bisogno di essere popolate dato che fungono da "ponte" tra più tabelle e hanno solo attributi che referenziano altre tabelle, mancando di esistenza autonoma.

UTENTE

```
INSERT INTO Utente (IP)
VALUES ('192.168.1.1'),
       ('192.168.1.2'),
       ('192.168.1.3'),
       ('192.168.1.4'),
       ('192.168.1.5'),
       ('192.168.1.6'),
       ('192.168.1.7'),
       ('192.168.1.8'),
       ('192.168.1.9'),
       ('192.168.1.10');
```

SPETTATORE REGISTRATO

```
INSERT INTO SpettatoreRegistrato (NomeUtente, IP, Password,
DataDiNascita, Email, NumeroDiTelefono)
VALUES
('JohnDoe', '192.168.1.1', 'password123', '1990-05-15',
'john.doe@example.com', '1234567890'),
('JaneDoe', '192.168.1.2', 'p@ssw0rd', '1985-09-20',
'jane.doe@example.com', '8697421035'),
('BabyDoe', '192.168.1.3', 'p@ssw0rd321', '2001-03-21',
'baby.doe@example.com', '2103458769'),
('AuloAgerio', '192.168.1.4', 'nomennescio', '1978-04-21',
'aulo.agerio@rm.com', '0753753753'),
('NumerioNegidio', '192.168.1.5', 'nomennescio', '1988-04-21',
'numerio.negidio@rm.com', '7537537530'),
('JanieDoe', '192.168.1.6', 'password98', '1995-09-20',
'janie.Doe@example.com', '1029384756'),
('JohnnyDoe', '192.168.1.7', 'p@ssw0rd', '1997-01-20',
'johnny.Doe@example.com', '5647382910');
```

SPETTATORE ANONIMO

```
INSERT INTO SpettatoreAnonimo (IP)
VALUES ('192.168.1.8'),
       ('192.168.1.9'),
       ('192.168.1.10');
```

STREAMER

```
INSERT INTO Streamer (NomeUtente, IP, MinutiTrasmessi,
MediaDiSpettatori, Calendario, NumeroDiLive, NumeroDiFollowers)
VALUES
('JohnDoe', '192.168.1.1', '70', '15', 'c1', '2', '78'),
```



```

('AuloAgerio', '192.168.1.4', '753', '21', 'c1', '4', '144'),
('JohnnyDoe', '192.168.1.7', '200', '17', 'c1', '1', '66');

```

CANALE

```

INSERT INTO Canale (NomeUtente, IP, MinutiTrasmessi,
MediaDiSpettatori, Calendario, NumeroDiLive, NumeroDiFollowers,
Facebook, Twitter, Instagram, YouTube)
VALUES
('JohnDoe', '192.168.1.1', '70', '15', 'c1', '2', '78', 'F1',
'T1', 'I1', 'Y1'),
('AuloAgerio', '192.168.1.4', '753', '21', 'c2', '4', '144', 'F2',
'T2', 'I2', 'Y2'),
('JohnnyDoe', '192.168.1.7', '200', '17', 'c3', '1', '66', 'F3',
'T3', 'I3', 'Y3');

```

LIVE

```

INSERT INTO Live (URL, Titolo, NumeroMedioDiSpettatori)
VALUES
('URL1', 'First Live', 23),
('URL2', 'Chill', 7),
('URL3', 'Roman Holiday', 21),
('URL4', 'Punic wars explained', 19),
('URL5', 'Cicerone vs Catilina: epic dissing', 27),
('URL6', '15 March: epic prank to Caesar', 21),
('URL7', 'First Live', 17);

```

VIDEO

```

INSERT INTO Video (URL, Titolo, NumeroMedioDiVisualizzazioniVideo,
Durata, Categorie, Tag)
VALUES
('URLV1', 'Begin of Diablo IV', 43, 32, Diablo IV, Fantasy),
('URLV6', 'Innocent prank', 33, 33, Prank, Betrayal),
('URLV7', 'Begin of Diablo IV', 23, 200, Diablo IV, Fantasy);

```

CLIP

```

INSERT INTO Clip (URL, Titolo, NumeroMedioDiVisualizzazioniClip)
VALUES
('URLC2', 'Boss fight', 53),
('URLC6', 'After Party', 45),
('URLC7', 'Boss fight', 43);

```

BITS

```

INSERT INTO Bits (SpettatoreRegistrato, PortafoglioVirtuale)
VALUES
('JohnDoe', 'p1'),
('JaneDoe', 'p2'),
('BabyDoe', 'p3'),
('AuloAgerio', 'p4'),

```

```
|| ('NumerioNegidio', 'p5'),  
|| ('JanieDoe', 'p6'),  
|| ('JohnnyDoe', 'p7');
```

TAGS

```
|| INSERT INTO Tags (Tag, Video)  
|| VALUES  
|| ('Italiano', 'URLV1'),  
|| ('Travel', 'URLV6'),  
|| ('Soundtrack', 'URLV7');
```

3.3 Operazioni

Di seguito sono riportate alcune delle operazioni di SQL per testare le funzionalita del DB.

OPERAZIONE 1

```
|| SELECT Titolo  
|| FROM Live  
|| WHERE Titolo = 'insert_a_title';
```

OPERAZIONE 2

```
|| SELECT NomeUtente  
|| FROM Streamer  
|| WHERE Titolo = 'insert_a_streamer_name';
```

OPERAZIONE 3

```
|| INSERT INTO Live (URL, Titolo, NumeroMedioDiSpettatori)  
|| VALUES ('url', 'title', avg_viewers);
```

OPERAZIONE 6

```
|| SELECT Nome, NumeroDiSubscriptions  
|| FROM Canale  
|| ORDER BY NumeroDiSubscriptions DESC;
```

OPERAZIONE 7

```
|| SELECT *  
|| FROM Streamer  
|| WHERE MinutiTrasmessi >= 500  
|| AND MediaDiSpettatori >= 3  
|| AND NumeroDiFollowers >= 50;
```

OPERAZIONE 8

```
|| SELECT Nome, MediaDiSpettatori  
|| FROM Canale  
|| ORDER BY MediaDiSpettatori DESC;
```