Laboratorio Basi di Dati 2022/2023

Andrea Garnerone - Corso A T2

---

1. Progettazzione Concettuale

1.1 Requisiti iniziali

Si vuole realizzare una base di dati per un servizio che permette di fare live streaming su vari argomenti. Il live streaming (o, più sinteticamente, la live) permette di interagire con il pubblico in tempo reale grazie a feed video, chat e altro.

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

Gli streamer hanno ciascuno un canale, che può essere caratterizzato tramite una descrizione. Per ogni canale, è possibile specificare una lista di social associati (ad esempio Instagram, YouTube, ecc.), un’immagine profilo e anche un trailer (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 follower), può diventare affiliate.

Le stream hanno degli orari. Ogni streamer ha un calendario in cui può dire quando farà stream e indicare il titolo delle prossime live.

I viewer possono diventare follower del canale degli streamer che preferiscono, e le loro preferenze sono raccolte in un elenco di followee a cui possono accedere dal loro profilo. I viewer possono inoltre supportare gli streamer tramite la subscription (a pagamento) al loro canale, ottenendo dei privilegi (emoticon personalizzate, ecc.). Inoltre, gli utenti hanno un portafoglio di bit (moneta virtuale che possono acquistare tramite la piattaforma), che possono usare per fare donazioni agli streamer.

Oltre a chattare pubblicamente, gli utenti possono scambiarsi messaggi privati.

La base di dati deve supportare le seguenti operazioni:

- Una volta al giorno si controllano le condizioni per la qualifica di affiliate.
- Una volta a settimana viene calcolata la classifica degli streamer più seguiti.

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

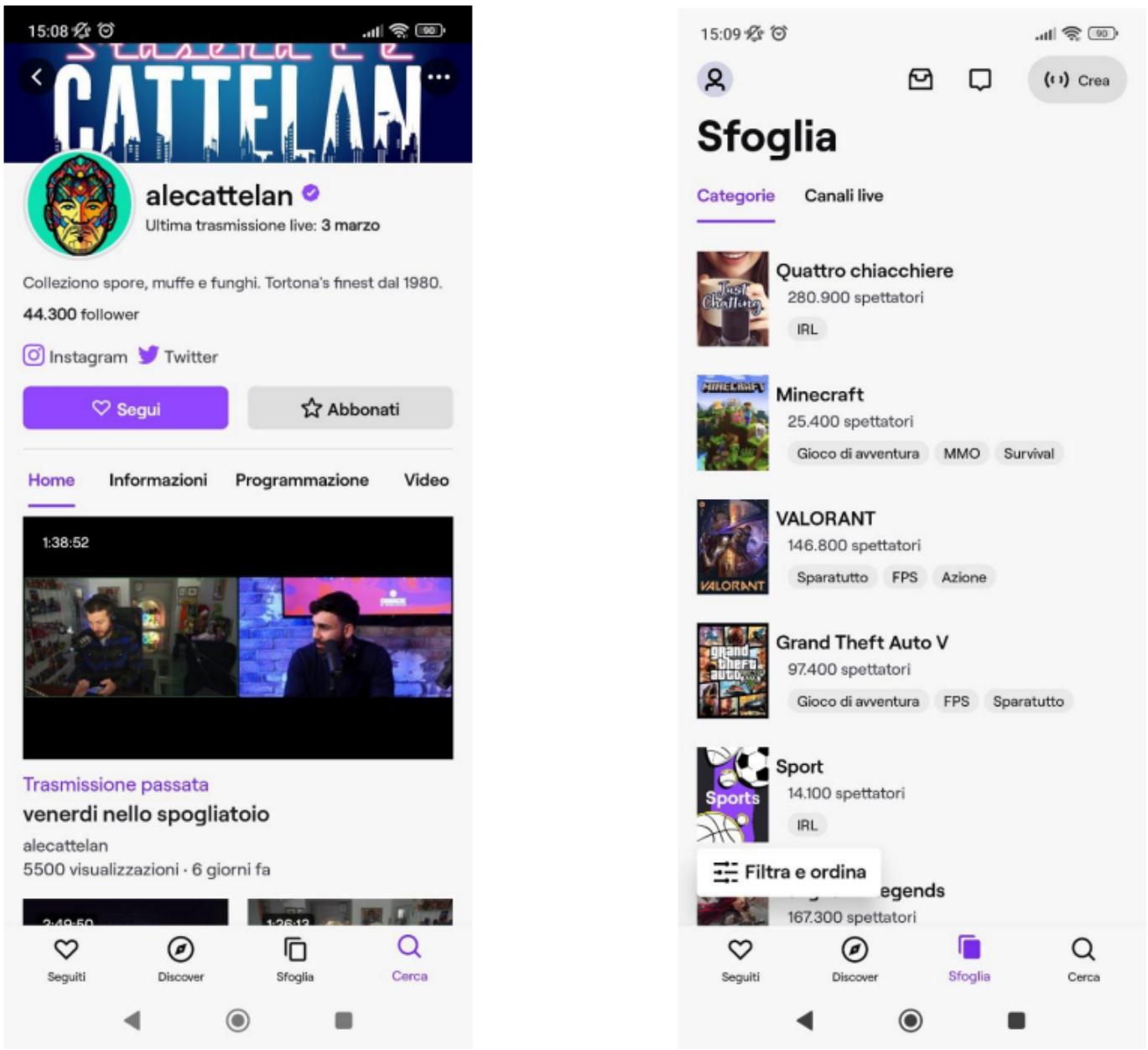


Figura 1 (a) La home del canale di alecattelan. (b) La lista delle categorie

1.2 Glossario dei Termini

Nello svolgimento del progetto abbiamo deciso di apportare qualche modifica ai requisiti inziali, laddove ci risultasse piu' corretto cambiare questi dati con delle informazioni piu' simili alla realtà, basate direttamente sulla piattaforma di riferimento Twitch:

- Invece che rappresentare "*I viewer possono diventare follower del canale degli streamer che preferiscono*", abbiamo rappresentato "*Gli utenti possono diventare follower di uno streamer*", e non solo gli spettatori.
- Invece che rappresentare "*Gli streamer hanno ciascuno un canale*" abbiamo rappresentato "*Tutti gli utenti hanno un canale*", e non solo gli streamer.
- Gli utenti possono donare solo agli streamer affiliate.
- Le clip sono estratte da una live.

Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Utente	La persona che usa la piattaforma per fruire di contenuti multimediali o fare live.	/	Streamer Spettatore Contenuti multimediali Canale Messaggio
Spettatore	Utente che guarda una live di uno streamer.	Viewer	Streamer Abbonamento
Streamer	Utente che produce contenuti multimendiali.	Creatore di contenuti	Utente Spettatore Live Canale Affiliate
Abbonamento	Uno spettatore che sostiene finanziariamente uno streamer. Gli abbonati possono usufruire di privilegi.	/	Spettatore Streamer
Affiliate	Programma proposto agli streamer che rispettano determinati requisiti.	/	Streamer
Canale	Il luogo virtuale dove uno streamer fà live.	/	Streamer Contenuti multimediali Calendario
Contenuti multimediali	Tutto ciò che un canale produce: Live, video (live passate) e clip (video di durata breve).	/	Canale Utente Live (Video, Clip)
Live	Trasmissioni in diretta dove gli streamer creano contenuti.	Live streaming Stream Dirette	Streamer Contenuti multimediali (Clip) Messaggio
Video	Conenuto multimediale riproducente una live passata.	/	Contenuti multimediali Live
Clip	Preve segmento di una live o di un video.	/	Contenuti multimediali
Calendario	Strumento che consente di pianificare le live future di un canale.	/	Canale
Messaggio	Testo inviato tramite tramite la chat in una live oppure tra utenti in privato.	/	Utente Live
Donazione	L’invio di bits, la moneta virtuale della piattaforma, da parte di un utente a uno streamer.	/	Utente Streamer

1.3 Requisiti rivisti

Frasi di carattere generale:

Per la base di dati rappresentiamo un servizio che permette di fare live su vari argomenti.

Per i contenuti multimediali memorizziamo solamente l’URL, assumendo che questi vengano gestiti da una piattaforma di video hosting e che quindi sia sufficiente memorizzare un URL.

Frasi relative agli utenti:

Per gli utenti rappresentiamo il fatto che possano essere spettatore o streamer, o entrambi.

Per la loro registrazione rappresentiamo nome utente, password, data di nascita, numero di telefono o indirizzo mail.

Una volta iscritti rappresentiamo il fatto che possono seguire lo streamer, fare live e chattare, sia tramite l’invio di messaggi privati tra utenti, che tramite quelli pubblici.

Inolter hanno un portafoglio di bit (moneta virtuale che possono acquistare tramite la piattaforma), che possono usare per fare donazioni agli streamer.

Frasi relative agli spettatori:

Per gli spettatori rappresentiamo il fatto che possono guardare le live in modo anonimo oppure essere registrati al servizio.

Per questi ultimi rappresentiamo l'elenco di canali seguiti, raccolti in un elenco di followee a cui possono accedere dal loro profilo. Inoltre possono supportare gli streamer tramite la subscription (a pagamento) al loro canale, ottenendo dei privilegi (emoticon personalizzate, ecc.)

Frasi relative agli streamer:

Per gli streamer rappresentiamo un canale, che puo' essere caratterizzato da una descrizione, e un calendario in cui può dire quando farà live e indicarne il titolo delle prossime.

Ogni steramer ha un numero di live effettuate, il numero di minuti trasmessi, il numero medio di spettatori simultanei.

Uno streamer che rispetta determinati parametri di performance (un minimo di 500 minuti trasmessi, una media di tre o più spettatori simultanei, almeno 50 follower) puo' diventare affiliate.

Frasi relative del canale:

Per un canale rappresentiamo una possibile lista di social associati (ad esempio Instagram, YouTube, ecc.), un'immagine profilo, una lista di followers e anche un trailer; eventuali live, video (live passate) e clip (video di durata breve).

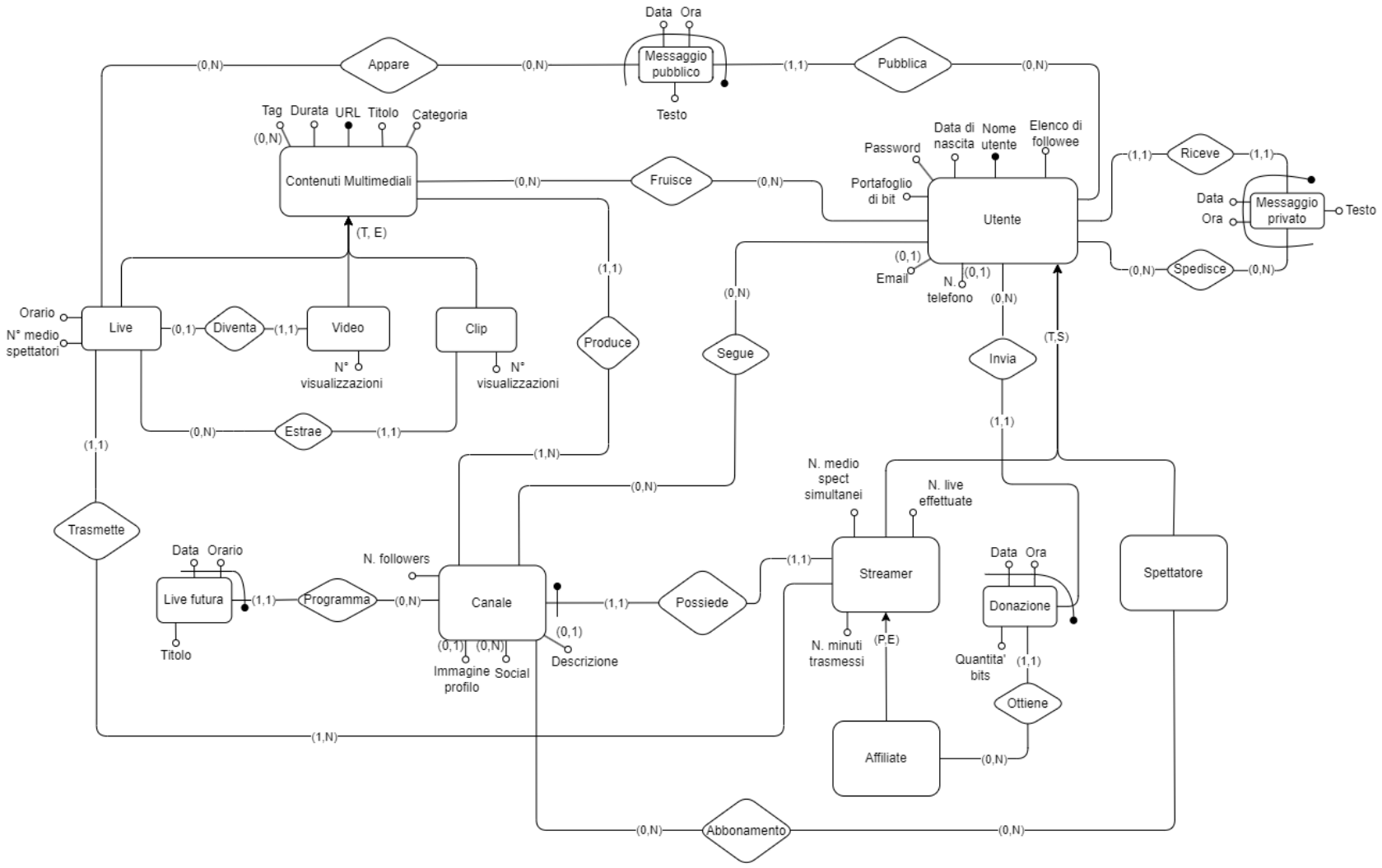
Frasi relative ai contenuti multimediali:

Per le live rappresentiamo la memorizzazione del numero medio di spettatori, del titolo, durata, categoria e orario.

Le live possono eventualmente diventare un video del canale.

Per i video rappresentiamo il numero di visualizzazioni, il titolo, la durata, la categoria ed eventuali tag associati.

1.4. Schema ER Principale



1.4.2 Business Rules

Integrità

- L’entità Utente si riferisce agli utenti registrati.
- LiveFutura.Orario > Live.Orario
- Per far parte dell’entità affiliate uno streamer deve avere: un minimo di 500 minuti trasmessi, una media di tre o più spettatori simultanei, almeno 50 follower.
- Utente.NumeroDiTelefono e Utente.Email sono facoltativi, ma deve essere specitificato un valore per almeno uno dei due.
- NomeUtente.Utente che *spedisce* MessaggioProvato ≠ NomeUtente.Utente che *riceve* MessaggioPrivato.
- Donazione.QuantitàBits > 0.
- Clip.Durata compresa tra 5 e 60 secondi. Se la durata è maggiore di 60 secondi allora è un video.

Derivazione

- Live.N.MedioSpettatori comprende nel conteggio anche le visualizzazioni degli utenti non registrati.

2. Progettazione Logica

2.1 Tavola dei volumi

Concetto	Tipo	Volume	Motivazione
Utente	E	60mln	Dato basato su stime relative al 2021
Spettatore	E	60mln	La maggior parte degli utenti e’ spettatore
Streamer	E	600.000	Secondo il principio del 90-9-1 solo l’1% degli utenti e’ streamer
Messaggio privato	E	45mln	Se ipotizziamo che solo il 5% degli utenti mandi messaggi privati, e che tra utenti ci sia una media di 30 messaggi scambiati
Messaggio pubblico	E	1.5mrd	Se consideriamo che circa il 50% degli utenti sia attivo in chat, consideriamo una media di 50 messaggi a utente
Donazione	E	1mln	Assumiamo che solo l’1% degli utenti doni con una media di 1-2 donazioni
Affiliate	E	300.000	Dato basato su stime relative al 2021
Live	E	200.000	Consideriamo che non tutti gli streamer siano in live
Video	E	6mln	Assumiamo che ci siano 10 video a canale
Clip	E	12mln	Assumiamo che ci siano 20 clip a canale
Contenuti multimediali	E	18.5mln	La somma dei contenuti multimediali (live, video e clip)
Canale	E	600.000	Uno per ogni streamer
Live futura	E	3mln	Se consideriamo che in media uno streamer ha 5 live pianificate
Spedisce	A	45mln	Il numero dei messaggi privati
Riceve	A	45mln	Il numero dei messaggi privati
Pubblica	A	1.5mrd	Il numero dei messaggi pubblici degli utenti
Appare	A	1.5mrd	Tutti i messaggi pubblicati appaiono in live
Invia	A	1mln	Se ci sono 1mln donazioni ci saranno 1mln invii
Ottiene	A	300.000	Una donazione per ogni streamer affiliate in media
Abbonamento	A	3mln	Non tutti gli spettatori sono abbonati a un canale
Fruisce	A	3.5mrd	Considerando che ogni utente abbia guardato in media 60 contenuti multimediali
Segue	A	300mln	Ogni utente segue in media 5 canali
Possiede	A	600.000	Un canale per streamer
Diventa	A	130.000	Ipotizziamo che 2/3 delle live diventino video
Estrae	A	1mln	Ipotizziamo che vengano estratte 5 clip a live
Programma	A	3mln	Una live futura ha un programma
Trasmette	A	200.000	Il numero delle live
Produce	A	18mln	Ogni canale produce una live, in media 10 video e 20 clip

2.2 Tavola delle operazioni

Operazione	Descrizione	Tipo	Frequenza	Motivazione
1	Fruizione di contenuti multimediali	I	50mln al giorno	L’operazione principale di una piattaforma di streaming e’ quella della visione di contenuti multimediali. (ci sono piu’ messaggi in live che spettatori)
2	Invia, da un utente, un messaggio pubblico in un una live	I	20mln al giorno	Una delle operazioni piu’ eseguite e’ quella dell’invio dei messaggi in una live, e sicuramente una delle piu’ dispendiose. (vengono fatte moltissime ricerche al giorno)
3	Cerca i dati del canale di uno streamer in base al nome	I	3mln al giorno	Operazione che simula la ricerca di un canale su Twitch
4	Segui, da utente, un canale di uno streamer	I	100.000 volte al giorno	Operazione che serve durante la visualizzaizone di un profilo utente, o della home page, nella quale devono apparire tutti i canali da lui seguiti.
5	Fai cominciare una live a uno streamer	I	100.000 al giorno	Operazione di modifica, usata frequentemente dagli streamer che iniziano una live.
6	Trasforma live in video	I	50.000 volte al giorno	Operazione che trasforma una live, quando finisce, in un video.
7	Estrai una clip da una live	I	50.000 volte al giorno	Operazione di creazione delle clip
8	Inserisci i dati di registrazione di un utente	I	15.000 volte al giorno	Operazione per la cerazione di nuovi utenti registrati alla piattaforma
9	Programma una live futura nel canale di uno streamer	I	10.000 volte al giorno	La funzione calendario
10	Trova a quali canali uno spettatore si e’ abbonato	I	500 volte al giorno	Operazione utile per il conteggio degli abbonati di un canale.
11	Dona a uno streamer da un’utente		5.000 volte al giorno	Operazione che serve per la donazione di bit.
12	Trova i messaggi privati inviati da un utente a un altro	I	100 volte al giorno	Operazione di lettura e non di scrittura in quanto altrimenti avrebbe una frequenza troppo alta, e falserebbe tutti i dati.
13	Trova i contenuti multimediali prodotti da ciascun canale	B	10 volte al giorno	Operazione che serve durante la visualizzazione di un canale, nella quale devono apparire tutte i contenuti da lui creati.
14	Controllo delle condizioni per la qulifica di affiliate	B	1 volta al giorno	Operazione inclusa nei requisiti inziali
15	Calcolo della classifica degli streamer più seguiti	B	1 volta a settimana	Operazione inclusa nei requisiti inziali

2.3 Restrutturazione dello Schema ER

2.3.1 Analisi delle Ridondanze

▼ L’attributo *Canale.N°followers*, la cui informazione e’ ricavabile da *Utente-segue-Canale*

Tavola dei volumi

Concetto	Tipo	Volume
Utente	E	60mln
Canale	E	600.000
Segue	A	300mln

Tavola delle operazioni

Operazione	Descrizione	Tipo	Frequenza
1	Cerca i dati del canale di uno streamer in base al nome	I	3mln al giorno
2	Segui, da utente, un canale di uno streamer	I	100.000 al giorno

Assenza di ridondanza

Operazione 1: Cerca i dati del canale di uno streamer in base al nome

Schema di operazione

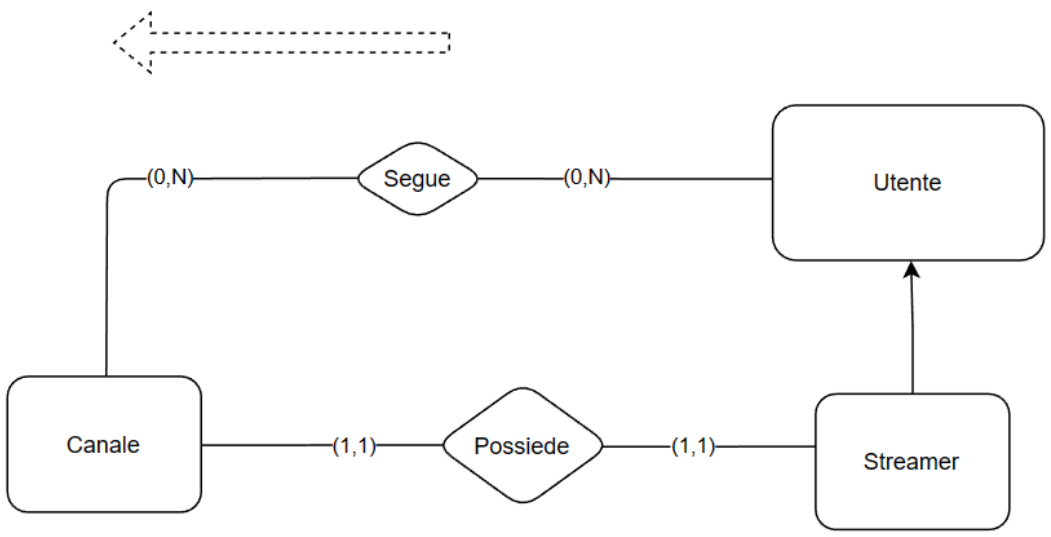


Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
Utente	Entità	2	L
Streamer	Entità	1	L
Canale	Entità	1	L
Segue	Associazione	1	L
Possiede	Associazione	1	L

Operazione 2: Segui, da utente, un canale di uno streamer

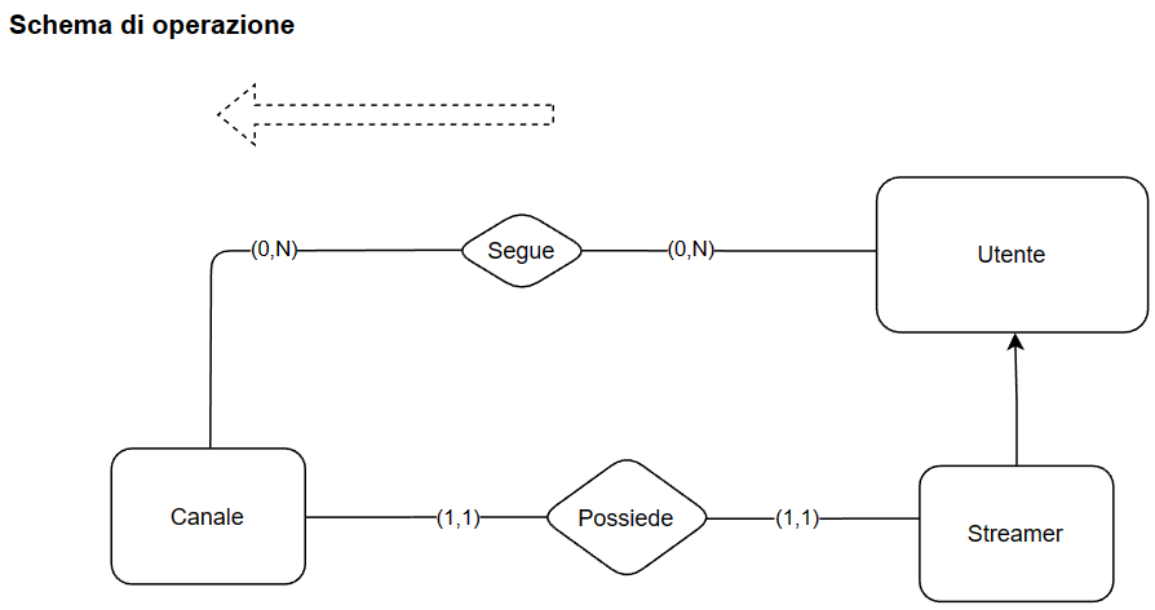


Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
Utente	Entità	2	S
Streamer	Entità	1	L
Canale	Entità	1	L
Segue	Associazione	1	S
Possiede	Associazione	1	L

Presenza di ridondanza

Operazione 1: Cerca i dati del canale di uno streamer in base al nome

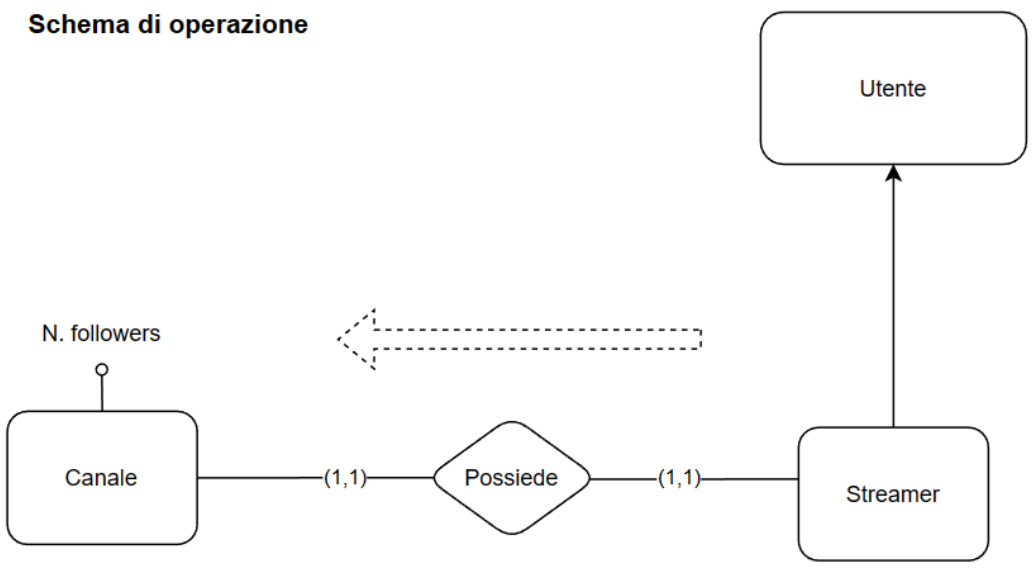


Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
Utente	Entità	1	L
Streamer	Entità	1	L

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

Operazione 2: Segui, da utente, un canale di uno streamer

Schema di operazione

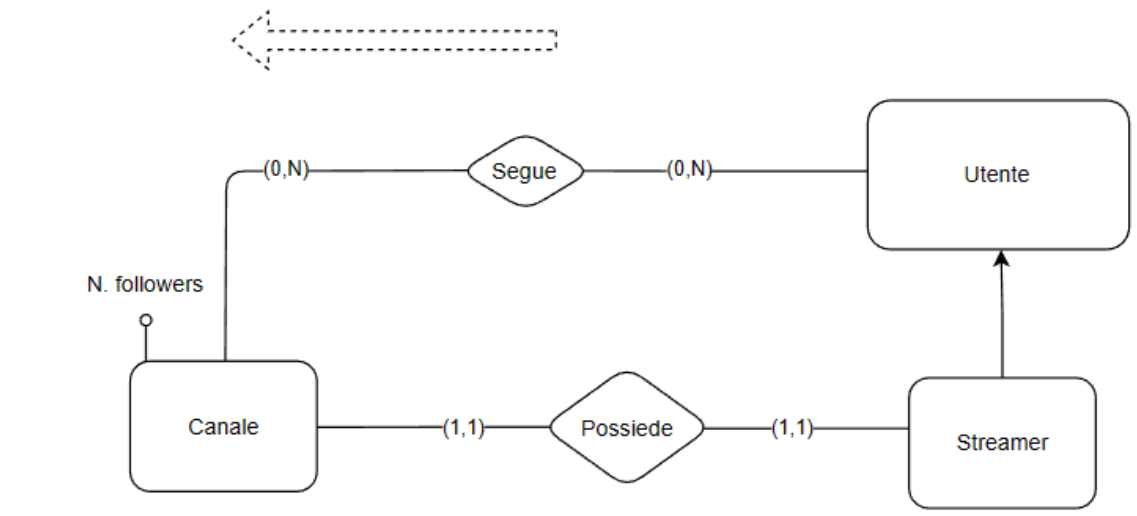


Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
Utente	Entità	2	S
Streamer	Entità	1	L
Canale	Entità	1	S
Segue	Associazione	1	S
Possiede	Associazione	1	L

Confronto di spazio e tempo occupato

Assenza di ridondanza

Spazio: 0 byte

Tempo:

- Operazione 1: 6 accessi * 3mln di volte al giorno = 18mln accessi al giorno
- Operazione 2: 9 accessi * 100.000 volte al giorno = 900.000 accessi al giorno
- Totale: 19mln di accessi al giorno

Presenza di ridondanza

Spazio: 4 byte * 600.000 = 2.400.000 (2,4 megabyte)

Tempo:

- Operazione 1: 4 accessi * 3mln di volte al giorno = 12mln accessi al giorno
- Operazione 2: 10 accessi * 100.000 volte al giorno = 1mln accessi al giorno
- Totale: 15mln di accessi al giorno

Conclusione:

Se decidiamo di eliminare la ridondanza, avremmo un risparmio di circa 2~3 MB, a fronte però di circa 4mln di accessi in più al giorno.

Altrimenti, se manteniamo la ridondanza, risparmiano circa 4mln di accessi al giorno a fronte di un costo di spazio di circa 2~3 MB

Decidiamo dunque di mantenere la ridondanza.

- L’associazione *Crea* che collega *Streamer* a *Live*. Infatti la stessa informazione e’ ricavabile passando da *Streamer-possiede-Canale, Canale-produce-ContenutiMultimediali, ContenutiMultimediali-Live*.

- L’attributo Video.N°visualizzazioni e Clip.N°visualizzazioni, dato che l’informazione che apportano e’ la stessa ricavabile dal percorso *Utente-fruisce-ContenutiMultimediali, ContenutiMultimediali-Video / ContenutiMultimediali-Clip*

2.3.2 Eliminazione delle generalizzazioni

La prima generalizzazione eliminata è quella relativa all’entità Utente, che ha come figli Streamer e Spettatore. In questo caso la decisione è quella di sostituire la generalizzazione con delle associazioni, dato che gli accessi ai figli sono nettamente separati rispetto a quelli al genitore. Ed essendo questa una delle considerazioni principali nella decisione di sostituire la generalizzazione con delle associazioni, la scelta è ricaduta sulla suddetta possibilità.
La seconda generalizzazione rimossa riguarda Streamer e Affiliate. Anche in questo caso, essendoci l’entità Donazione che collega Affiliate a Utente tramite due associazioni, gli accessi a Streamer e Affiliate avvengono separatamente, quindi è bene mantenere le due entità in vita, senza accorpare nessuna delle due all’altra. Di conseguenza, come per la generalizzazione precedente, le due entità sono state collegate da un’associazione.
Discorso diverso per l’ultima generalizzazione, che riguarda Contenuti Multimediali. In questo caso si è notato che questa entità si poteva eliminare, lasciando invariato il funzionamento dello schema. Quindi la soluzione adottata è quella dell’accorpamento del genitore nei figli. Di conseguenza anche tutti gli attributi di Contenuti Multimediali sono stati riportati per tutti i figli, così come le associazioni collegate al padre. Questa soluzione si è resa possibile anche grazie al fatto che la generalizzazione è totale, requisito necessario per l’accorpamento, e dal fatto che gli accessi alle entità figlie sono distinte, come richiesto dalle slides del corso.

2.3.3 Partizionamento/accorpamento di entità

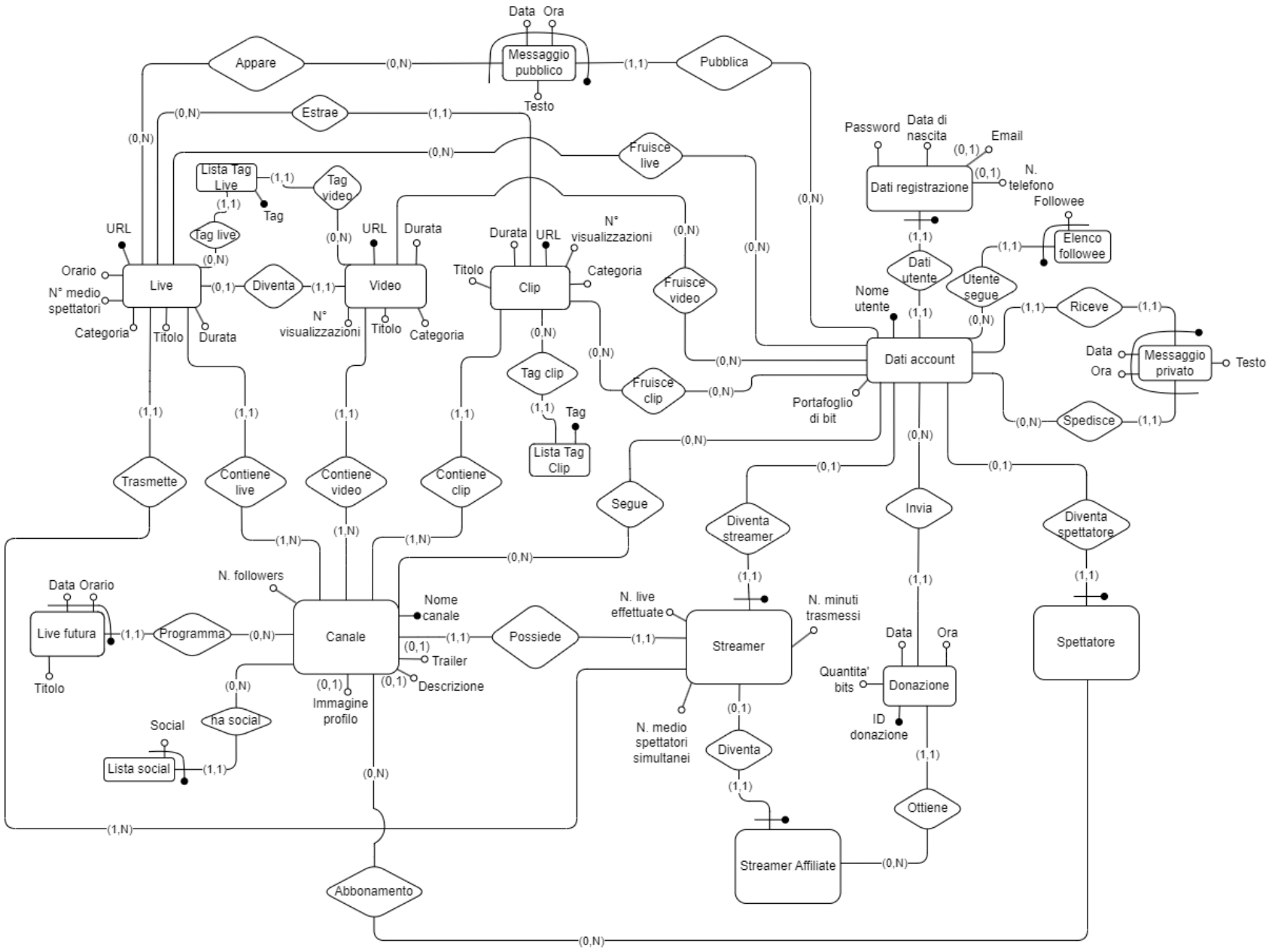
L’unico partizionamento riguarda l'entità Utente. Infatti, gli attributi di questa si possono dividere in due categorie: quelli relativi ai dati di registrazione e quelli relativi all'account. Fatta questa distinzione, la soluzione adottata è quella del partizionamento di Utente in due entità, una relativa ai dati della registrazione e l'altra relativa ai dati dell'account. Si noti come il Nome utente sia nell'entità relativa all'account e non nei dati di registrazione, come verrebbe spontaneo. Questo perché il nome utente di un account è sì un dato di registrazione, ma viene anche visualizzato nel profilo, quindi nell'account, di un utente. Dunque, data la possibilità di inserirlo in entrambe le entità, ha senso che sia nell'entità dei dati dell'account, dato che questa entità viene usata molto più spesso rispetto all'altra, e dunque risulta meno dispendioso in questo modo. Un’altra particolarità è che solo l’entità Dati accounti è stata ricollegata con tutte le associazioni di Utente. Questo perchè i dati dell’account di un utente sono privati, e non condivisi con gli altri utenti. Inoltre si può considerare Dati accounti come l’entità con peso maggiore, e quella più simile a Utente. Dunque l’unica collegata al resto dello schema.

Non è stato fatto nessun accorpamento.

2.3.4 Identificatori principali

Nell’entità Donazione l’identificatore scelto nello schema ER è esterno con Utente. Tuttavia questo è molto dispendioso, dunque è stato inserito un nuovo identificatore, ID donazione, che identifica univocamente una donazione grazie a un ID associato a ogni donazione. In tutti gli altri casi non è sembrato necessario introdurre un identificatore surrogato. Per esempio Live futura mantiene l’identificatore esterno perché l’entità è poco usata dal database, dunque non ci sono riduzioni considerevoli di prestazioni nel database.

2.4 Schema E-R Ristrutturato



1.4.1 Business Rules

Integrità

- Le entità Dati registrazione e Dati accounti si riferiscono solamente agli utenti registrati.
- L’associazione Diventa viene usata solo se uno Streamer ha: Steramer.MinutiTrasmessi ≥ 500, Steramer.N.MedioSpettatoriSimultanei ≥ 3, Streamer.Follower ≥ 50.
- DatiRegistrazione.NumeroDiTelefono e DatiRegistrazione.Email sono facoltativi, ma deve essere specitificato un valore per almeno uno dei due.
- NomeUtente.Utente che *spedisce* MessaggioProvato ≠ NomeUtente.Utente che *riceve* MessaggioPrivato.
- Donazione.QuantitàBits > 0.
- Clip.Durata compresa tra 5 e 60 secondi. Se la durata è maggiore di 60 secondi allora è un video.

Derivazione

- Live.N.MedioSpettatori comprende nel conteggio anche le visualizzazioni degli utenti non registrati.

2.5 Schema Relazionale

Dati registrazione(NomeUtente, Password, Data di nascita, Email, N. telefono)

Dati account(Nome Utente, Portafoglio di bit)

Messaggio privato(Data, Ora, NomeMittente, NomeRicevente, Testo)

Spettatore(NomeUtente)

Elenco followee(Followee, Utente)

Donazione(ID donazione, NomeUtenteAccount, NomeStreamer Data, Ora, Quantità di bits)

Streamer(NomeUtente, N. minuti trasmessi, N, live effettuate, N. medio spettatori simultanei)

Streamer Affiliate(NomeUtente)

Messaggio pubblico(Data, Ora, NomeMittente, ChatLive, Testo)

Live(URL, Durata, Titolo, Categoria, N. medio spettatori, Orario, Streamer, Video*, Canale)

Appare(Live, Utente)

Fruisce live(Live, Utente)

Lista Tag Live(Tag, Live, Video)

Clip(URL, Live, Durata, Titolo, Categoria, N. visualizzazioni, Canale)

Fruisce clip(Clip, Utente)

Lista Tag Clip(Tag, Clip)

Video(URL, Durata, Categoria, Titolo, N. visualizzazioni, Canale)

Fruisce video(Video, Utente)

Canale(Nome canale, Trailer, Descrizione, Immagine progilo, N. followers, Streamer)

Segue(Utente, Canale)

Abbonamento(Spettatore, Canale)

Live futura(Data, Orario, Canale)

Lista social(Social, Canale)

Vincoli

- Dati registrazione(NomeUtente) referenzia Dati account(Nome utente)
- Messaggio privato(NomeMittente) referenzia Dati account(Nome utente)
- Messaggio privato(NomeRicevente) referenzia Dati account(Nome utente)
- Spettatore(NomeUtente) referenzia Dati account(Nome Utente)
- Elenco followee(Utente) referenzia Dati utente(Nome utente)
- Donazione(NomeUtenteAccount) referenzia Dati account(Nome utente)
- Donazione(NomeStreamer) referenzia Steramer Affiliate(NomeUtente)
- Streamer(NomeUtente) referenzia Dati account(Nome utente)

- Streamer(NomeUtente) unique
- Streame Affiliate(NomeUtente) referenzia Streamer(NomeUtente)
- Messaggio pubblico(NomeMittente) referenzia Dati account(Nome utente)
- Messaggio pubblico(ChatLive) referenzia Live(URL)
- Fruisce live(Live) referenzia Live(URL)
- Fruisce live(Utente) referenzia Dati utente (Nome utente)
- Appare(Live) referenzia Live(URL)
- Appare(Utente) referenzia Dati utente (Nome utente)
- Lista tag Live(Live) referenzia Live(URL)
- Lista tag Live(Video) referenzia Video(URL)
- Live(Streamer) referenzia Streamer(NomeUtente)
- Live(video) referenzia Video(URL)
- Live(video) unique
- Live(Canale) referenzia Canale(Nome canale)
- Clip(Live) referenzia Live(URL)
- Clip(Canale) referenzia Canale(Nome canale)
- Fruisce clip(Clip) referenzia Clip(URL)
- Fruisce clip(Live) referenzia Dati utente(Nome utente)
- Lista tag clip(Clip) referenzia Clip(URL)
- Fruisce video(Video) referenzia Video(URL)
- Fruisce video(Utente) referenzia Dati utente(Nome utente)
- Video(Canale) referenzia Canale(Nome canale)
- Canale(Streamer) referenzia Streamer(NomeUtente)
- Segue(Utente) referenzia Dati utente(Nome utente)
- Segue(Canale) referenzia Canale(Nome canale)
- Abbonamento(Spettatore) referenzia Spettatore(NomeUtente)
- Abbonamento(Canale) referenzia Canale(Nome canale)
- Live futura(Canale) referenzia Canale(Nome canale)
- Lista social(Canale) referenzia Canale(Nome canale)

3. Implementazione

3.1 DDL di creazione del database

```
begin;

create table DatiAccount (
  NomeUtente varchar(25) primary key,
  Bits integer
);

create table DatiRegistrazione (
  NomeUtente varchar(25) primary key,
  Password varchar(25) not null,
  DataNascita date not null,
  Emai varchar(50),
  Telefono varchar(12),
  foreign key (NomeUtente) references DatiAccount(NomeUtente)
      on update cascade
      on delete no action
);

create table MessaggioPrivato (
  Data_e_Ora timestamp,
  NomeMittente varchar(25),
  NomeRicevente varchar(25),
  Testo varchar(1000) not null,
  primary key (Data_e_Ora, NomeMittente, NomeRicevente),
  foreign key (NomeMittente) references DatiAccount(NomeUtente)
      on update cascade
      on delete no action,
  foreign key (NomeRicevente) references DatiAccount(NomeUtente)
      on update cascade
      on delete no action
);

create table Spettatore (
  NomeUtente varchar(25) primary key,
  foreign key (NomeUtente) references DatiAccount(NomeUtente)
      on update cascade
      on delete no action
);

create table ElencoFollowee (
  Utente varchar(25),
  Followee varchar(25),
  primary key (Utente, Followee),
  foreign key (Utente) references DatiAccount(NomeUtente)
      on update cascade
      on delete no action
);

create table Streamer (
  NomeUtente varchar(25) primary key,
  MinutiTrasmessi interval day to second(2), --not null,
  LiveEffettuate numeric(3) not null,
  NSpectSimul integer, --not null,
  foreign key (NomeUtente) references DatiAccount(NomeUtente)
      on update cascade
      on delete no action
);

create table StreamerAffiliate (
  NomeStreamer varchar(25) primary key,
  foreign key (NomeStreamer) references DatiAccount(NomeUtente)
      on update cascade
      on delete no action
);

create table Donazione (
  ID numeric(7) primary key,
  NomeUtenteAccount varchar(25) not null,
  NomeStreamer varchar(25) not null,
  Data_e_Ora timestamp not null,
  Bits numeric(5) not null,
```

```
  foreign key (NomeUtenteAccount) references DatiAccount(NomeUtente)
    on update cascade
    on delete no action,
  foreign key (NomeStreamer) references StreamerAffiliate(NomeStreamer)
    on update cascade
    on delete no action
);

create table Canale (
  NomeCanale varchar(25) primary key,
  Trailer bytea,
  Descrizione text,
  ImmagineProfilo bytea,
  NFollowers integer not null,
  Streamer varchar(25) not null,
  foreign key (Streamer) references Streamer(NomeUtente)
      on update cascade
      on delete no action
);

create table Video (
  URL varchar(1000) primary key,
  Durata interval day to second not null,
  Titolo varchar(500)  not null,
  Categoria varchar(100),
  NVisual integer not null,
  Canale varchar(25) not null,
  foreign key (Canale) references Canale(NomeCanale)
);

create table Live (
  URL varchar(1000) primary key,
  Durata interval day to second(2),
  Titolo varchar(500) not null,
  Categoria varchar(100),
  NMedioSpect smallint,
  Orario timestamp(2),
  Streamer varchar(25) not null,
  Video varchar(1000) unique,
  Canale varchar(25) not null,
  foreign key (Streamer) references Streamer(NomeUtente)
      on update cascade
      on delete no action,
  foreign key (Video) references Video(URL)
      on update cascade
      on delete no action,
  foreign key (Canale) references Canale(NomeCanale)
      on update cascade
      on delete no action
);


create table MessaggioPubblico (
  Data_e_Ora timestamp,
  NomeMittente varchar(25),
  ChatLive varchar(1000),
  Testo varchar(1000) not null,
  primary key (Data_e_Ora, NomeMittente, ChatLive),
  foreign key (NomeMittente) references DatiAccount(NomeUtente)
      on update cascade
      on delete no action,
  foreign key (ChatLive) references Live(URL)
      on update cascade
      on delete no action
);

create table Appare (
  Live varchar(1000),
  Utente varchar(20),
  primary key (Live, Utente),
  foreign key (Live) references Live(URL)
      on update cascade
      on delete no action,
  foreign key (Utente) references DatiAccount(NomeUtente)
      on update cascade
      on delete no action
);
```

```
create table FruisceLive (
  Live varchar(1000),
  Utente varchar(20),
  primary key (Live, Utente),
  foreign key (Live) references Live(URL)
      on update cascade
      on delete no action,
  foreign key (Utente) references DatiAccount(NomeUtente)
      on update cascade
      on delete no action
);

create table ListaTagLive (
  Tag varchar(30),
  Live varchar(1000),
  Video varchar(1000),
  primary key (Tag, Live),
  foreign key (Live) references Live(URL)
      on update cascade
      on delete no action,
  foreign key (Video) references Video(URL)
      on update cascade
      on delete no action
);

create table Clip (
  URL varchar(1000) primary key,
  Durata interval day to second(2),
  Titolo varchar(140)  not null,
  Categoria varchar(100),
  Live varchar(1000) not null,
  NVisual integer not null,
  Canale varchar(25) not null,
  foreign key (Live) references Live(URL)
      on update cascade
      on delete no action,
  foreign key (Canale) references Canale(NomeCanale)
      on update cascade
      on delete no action
);

create table FruisceClip (
  Clip varchar(1000),
  Utente varchar(25),
  primary key (Clip, Utente),
  foreign key (Clip) references Clip(URL)
      on update cascade
      on delete no action,
  foreign key (Utente) references DatiAccount(NomeUtente)
      on update cascade
      on delete no action
);

create table ListaTagClip (
  Tag varchar(30),
  Clip varchar(1000),
  primary key (Tag, Clip),
  foreign key (Clip) references Clip(URL)
      on update cascade
      on delete no action
);

create table FruisceVideo (
  Video varchar(1000),
  Utente varchar(25),
  primary key (Video, Utente),
  foreign key (Video) references Video(URL)
      on update cascade
      on delete no action,
  foreign key (Utente) references DatiAccount(NomeUtente)
      on update cascade
      on delete no action
);

create table Segue (
  Utente varchar(25),
  Canale varchar(25),
  primary key (Utente, Canale),
  foreign key (Utente) references DatiAccount(NomeUtente)
```

```
      on update cascade
      on delete no action,
  foreign key (Canale) references Canale(NomeCanale)
      on update cascade
      on delete no action
);

create table Abbonamento (
  Spettatore varchar(25),
  Canale varchar(25),
  primary key (Spettatore, Canale),
  foreign key (Spettatore) references Spettatore(NomeUtente)
      on update cascade
      on delete no action,
  foreign key (Canale) references Canale(NomeCanale)
      on update cascade
      on delete no action
);

create table LiveFutura (
  Data_e_Ora timestamp,
  Canale varchar(25),
  primary key (Data_e_Ora, Canale),
  foreign key (Canale) references Canale(NomeCanale)
      on update cascade
      on delete no action
);

create table ListaSocial (
  Social varchar(15),
  Canale varchar(25),
  primary key (Social, Canale),
  foreign key (Canale) references Canale(NomeCanale)
      on update cascade
      on delete no action
);


commit;
```

3.2 DML di Popolamento di Tutte le Tabelle del Database

```
insert into DatiAccount values ('Dario Moccia', 0);
insert into DatiAccount values ('Enkk', 0);
insert into DatiAccount values ('Voghelita', 0);
insert into DatiAccount values ('NanniTwitch', 0);
insert into DatiAccount values ('Michelle Puttini', 0);
insert into DatiAccount values ('StudyTme', 0);
insert into DatiAccount values ('cavernadiplatone', 0);
insert into DatiAccount values ('pescanor', 0);
insert into DatiAccount values ('lynch227', 100);
insert into DatiAccount values ('agneseinnocente', 0);
insert into DatiAccount values ('loremosca', 10000);
insert into DatiAccount values ('sestroverso', 150);

insert into DatiRegistrazione(NomeUtente, Password, DataNascita) values
                             ('Dario Moccia', 'password', '1990-09-10');
insert into DatiRegistrazione(NomeUtente, Password, DataNascita) values
                             ('Enkk', 'asdfgh', '1987-02-21');
insert into DatiRegistrazione(NomeUtente, Password, DataNascita) values
                             ('Voghelita', '123456', '1995-11-13');
insert into DatiRegistrazione(NomeUtente, Password, DataNascita) values
                             ('NanniTwitch', 'qwerty', '1992-05-30');
insert into DatiRegistrazione(NomeUtente, Password, DataNascita) values
                             ('Michelle Puttini', '1qaz2wsx', '1994-07-14');
insert into DatiRegistrazione(NomeUtente, Password, DataNascita) values
                             ('StudyTme', '987654321', '1995-08-17');
insert into DatiRegistrazione(NomeUtente, Password, DataNascita) values
                             ('cavernadiplatone', 'zxcvbn', '1988-12-05');

insert into Spettatore values ('pescanor');
insert into Spettatore values ('lynch227');
insert into Spettatore values ('agneseinnocente');
```

```
insert into Spettatore values ('loremosca');
insert into Spettatore values ('sestroverso');

insert into Streamer (NomeUtente, LiveEffettuate) values ('Dario Moccia', 57);
insert into Streamer (NomeUtente, LiveEffettuate) values ('Voghelita', 4);
insert into Streamer (NomeUtente, LiveEffettuate) values ('cavernadiplatone', 45);
insert into Streamer (NomeUtente, LiveEffettuate) values ('NanniTwitch', 50);
insert into Streamer (NomeUtente, LiveEffettuate) values ('Enkk', 12);
insert into Streamer (NomeUtente, LiveEffettuate) values ('Michelle Puttini', 41);
insert into Streamer (NomeUtente, LiveEffettuate) values ('StudyTme', 51);

insert into Canale (NomeCanale, NFollowers, Streamer) values ('NanniTwitch', 87206, 'NanniTwitch');
insert into Canale (NomeCanale, NFollowers, Streamer) values ('Dario Moccia', 443161, 'Dario Moccia');
insert into Canale (NomeCanale, NFollowers, Streamer) values ('Enkk', 165901, 'Enkk');
insert into Canale (NomeCanale, NFollowers, Streamer) values ('Voghelita', 83641, 'Voghelita');
insert into Canale (NomeCanale, NFollowers, Streamer) values ('Michelle Puttini', 195202, 'Michelle Puttini');
insert into Canale (NomeCanale, NFollowers, Streamer) values ('StudyTme', 97976, 'StudyTme');

insert into Segue values ('pescanor', 'Dario Moccia');
insert into Segue values ('agneseinnocente', 'Dario Moccia');
insert into Segue values ('lynch227', 'Enkk');
insert into Segue values ('Voghelita', 'Michelle Puttini');
insert into Segue values ('lynch227', 'StudyTme');

insert into StreamerAffiliate values ('Dario Moccia');
insert into StreamerAffiliate values ('Voghelita');
insert into StreamerAffiliate values ('Enkk');
insert into StreamerAffiliate values ('StudyTme');

insert into Donazione values (1, 'sestroverso', 'StudyTme', '2023-06-04 11:05:06', 100);

  insert into Video (URL, Durata, Titolo, Categoria, NVisual, Canale) values
         ('https://www.twitch.tv/videos/1833426755', '0-07:55:15', 'Intervista a Filippo Giardina |
         Alle 22:45 Cinema Cult con Lo Squalo | 23:45 Zelda !prime !telegram !publive',
         'Just chatting', 71378, 'Dario Moccia');

insert into Abbonamento values ('agneseinnocente', 'Dario Moccia');

insert into ElencoFollowee values ('lynch227', 'Enkk');
insert into ElencoFollowee values ('lynch227', 'StudyTme');
```

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

```
delete from Streamer where NomeUtente = 'NanniTwitch';  --Operazione non permessa
                                                        --dal database
delete from StreamerAffiliate where NomeStreamer = 'Enkk';

drop table Canale; --Operazione non permessa dal database

update DatiAccount
set NomeUtente = 'Caverna di platone'
where NomeUtente = 'cavernadiplatone';

delete from Streamer where NomeUtente = 'Caverna di platone';
delete from DatiRegistrazione where NomeUtente = 'Caverna di platone';
delete from DatiAccount where NomeUtente = 'Caverna di platone';
```