

UNIVERSITÀ DEGLI STUDI DI TORINO



BASI DI DATI

MFN0602

Progetto

Oskar Heise

2022/2023

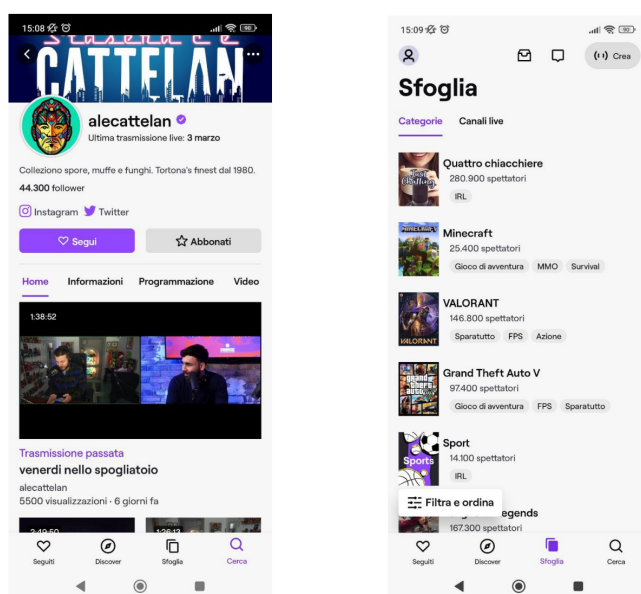
Indice

1	Progettazione Concettuale	2
1.1	Requisiti iniziali	2
1.2	Glossario dei termini	3
1.3	Requisiti rivisti e strutturati in frasi omogenee	4
1.4	Schema E-R principale + business rules	5
1.4.1	Schema E-R	5
1.4.2	Business Rules	6
1.4.3	Vincoli di Integrità	8
2	Progettazione Logica	9
2.1	Tavola dei volumi	9
2.2	Tavola dei volumi	9
2.3	Spiegazione dei dati	10
2.4	Tavola delle operazioni	11
2.5	Ristrutturazione dello schema E-R	11
2.5.1	Analisi delle ridondanze	11
2.5.2	Eliminazione delle generalizzazioni	15
2.5.3	Eventuale partizionamento/accorpamento di entità e associazioni	17
2.5.4	Eventuale eliminazione degli attributi composti e degli attributi multivalore	17
2.5.5	Eventuale scelta degli identificatori principali	17
2.6	Schema E-R ristrutturato + business rules	18
2.6.1	Schema E-R	18
2.7	Schema relazionale	19
3	Implementazione	20
3.1	DDL di creazione del database	20
3.2	DML di popolamento di tutte le tabelle del database	23
3.3	Qualche operazione di cancellazione e modifica per verificare i vincoli e gli effetti causati da operazioni su chiavi esterne	27
4	Extra	27

1 Progettazione Concettuale

1.1 Requisiti iniziali

Si vuole realizzare una base di dati per un servizio che permette di fare live streaming su vari argomenti. Il live streaming (o, più sinteticamente, la live) permette di interagire con il pubblico in tempo reale grazie a feed video, chat e altro. Ogni utente può essere spettatore o streamer, o entrambi. Gli spettatori possono essere registrati al servizio oppure possono guardare le live in modo anonimo. Per registrarsi, gli utenti devono indicare nome utente, password, data di nascita, numero di telefono o indirizzo mail. Gli utenti iscritti possono chattare, seguire lo streamer, creare dirette. Gli streamer hanno ciascuno un canale, che può essere caratterizzato tramite una descrizione. Per ogni canale, è possibile specificare una lista di social associati (ad esempio Instagram, YouTube, ecc.), un'immagine profilo e anche un trailer (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 follower a cui possono accedere dal loro profilo. I viewer possono inoltre supportare gli streamer tramite la subscription (a pagamento) al loro canale, ottenendo dei privilegi (emoticon personalizzate, ecc.). Inoltre, gli utenti hanno un portafoglio di bit (moneta virtuale che possono acquistare tramite la piattaforma), che possono usare per fare donazioni agli streamer. Oltre a chattare pubblicamente, gli utenti possono scambiarsi messaggi privati.

La base di dati deve supportare le seguenti operazioni:

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

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

1.2 Glossario dei termini

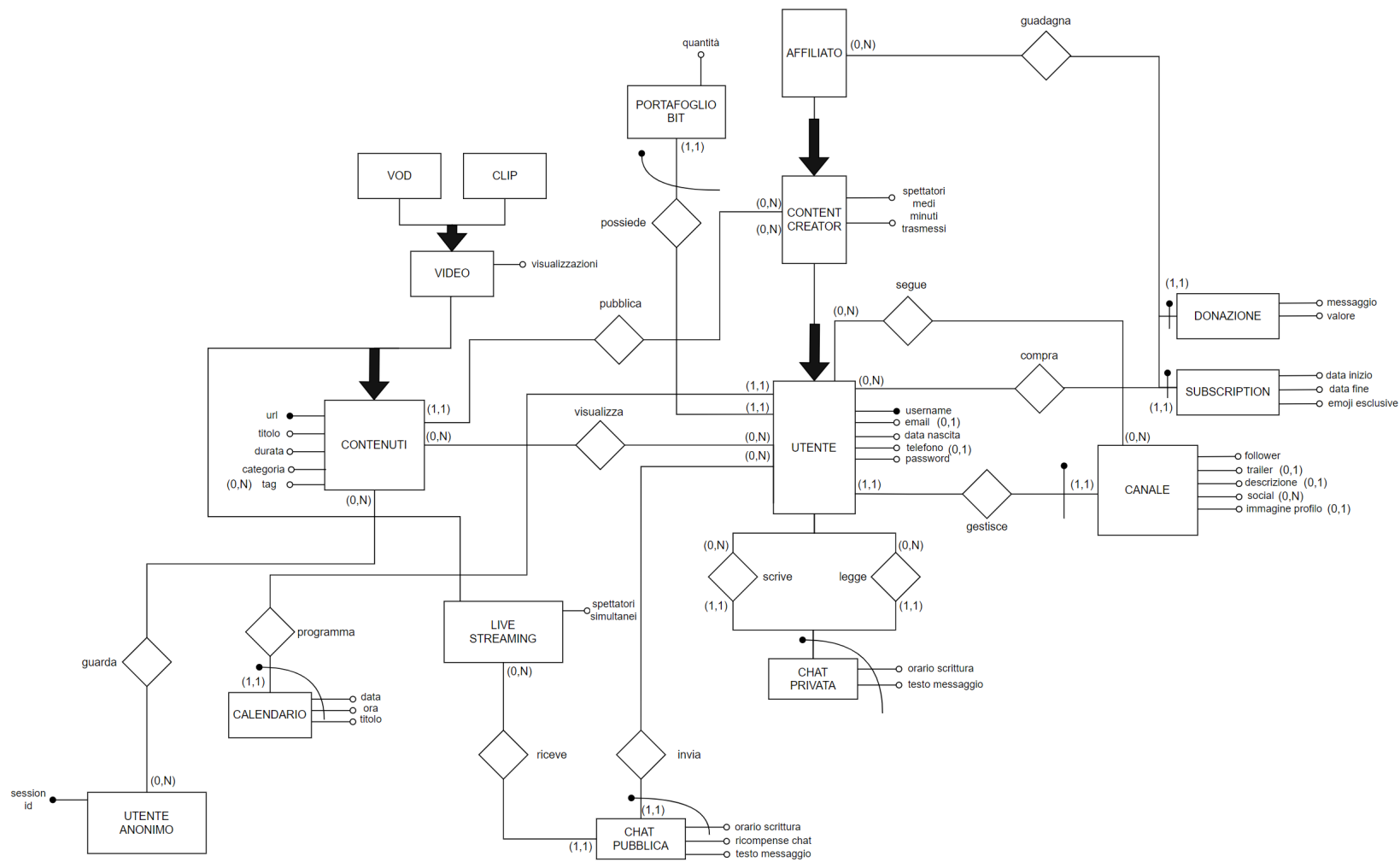
Termine	Descrizione	Sinonimi	Collegamenti
Utente	Partecipante della piattaforma	Pubblico	Calendario, Portafoglio bit, Chat pubblica, Chat privata, Canale, Subscription, Donazione
Content creator	Creatore di contenuti	Streamer, Creatore di contenuti	Contenuti
Affiliato	Partner Twitch	N.D.	Donazione, Subscription
Portafoglio bit	Borsellino virtuale di bit	Wallet	Utente
VOD	Video on Demand	N.D.	N.D.
Clip	Momenti salienti di un VOD	Video di breve durata	N.D.
Video	Contenuto video	Live passata	N.D.
Contenuti	Materiale contenutistico	N.D.	Content Creator
Live streaming	Trasmissione in diretta	Live, Stream	Utente anonimo, Chat pubblica, Content creator
Calendario	Programma settimanale	N.D.	Live streaming, Utente
Utente anonimo	Anonimo	Viewer	Live streaming
Chat pubblica	Chatroom globale	Chat	Utente, Live streaming
Chat privata	Conversazione privata	Chat	Utente
Canale	Profilo utente	N.D.	Utente
Subscription	Iscrizione	N.D.	Utente, Affiliato
Donazione	Contributo monetario	N.D.	Utente, Affiliato

1.3 Requisiti rivisti e strutturati in frasi omogenee

- La piattaforma per la quale si vuole realizzare una base di dati è destinata ad ospitare contenuti multimediali, in particolare live streaming, video e clip. Il live streaming 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. Per registrarsi, gli utenti devono indicare nome utente, password, data di nascita, numero di telefono o indirizzo mail. Gli utenti registrati possono chattare, seguire lo streamer, creare live streaming.
- Gli spettatori possono essere registrati al servizio oppure possono guardare le live streaming in modo anonimo. Gli spettatori registrati 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. Gli spettatori 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 registrati possono scambiarsi messaggi privati.
- Per ogni streamer si memorizzano il numero di live streaming effettuate, il numero di minuti trasmessi, il numero di medio di spettatori simultanei. 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*. Ogni streamer ha un calendario in cui può dire quando farà stream e indicare il titolo delle prossime live streaming.
- Gli streamer hanno ciascuno un canale, che può essere caratterizzato tramite una descrizione. In ogni canale possono esserci live streaming, video (live streaming passate) e clip (video di durata breve). Ognuno ha un titolo, una durata, appartiene a una categoria e può essere associato a diversi tag. Per ogni live streaming viene memorizzato il numero medio di spettatori mentre per i video e le clip il numero di visualizzazioni. Le live streaming possono anche non diventare video del canale. 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 Schema E-R principale + business rules

1.4.1 Schema E-R



1.4.2 Business Rules

Dizionario dei dati per le entità

Entità	Descrizione	Attributi	Identificatore
Utente	Partecipante della piattaforma	Username, Email, Data di nascita, Telefono, Password	Username
Content creator	Creatore di contenuti	Username, Email, Data di nascita, Telefono, Password, Spettatori medi, Minuti trasmessi	Username
Affiliato	Partner Twitch	Username, Email, Data di nascita, Telefono, Password, Spettatori medi, Minuti trasmessi	Username
Portafoglio bit	Borsellino virtuale di bit	Quantità	Username
Contenuti	Materiale contenutistico	URL, Titolo, Durata, Categoria, Tag	URL
Video	Contenuto video	URL, Visualizzazioni, Titolo, Durata, Categoria, Tag	URL
VOD	Video on Demand	URL, Visualizzazioni, Titolo, Durata, Categoria, Tag	URL
Clip	Momenti salienti di una diretta	URL, Visualizzazioni, Titolo, Durata, Categoria, Tag	URL
Live streaming	Trasmissione in diretta	URL, Spettatori simultanei, Titolo, Durata, Categoria, Tag	URL
Calendario	Programma settimanale	Data, Ora, Titolo	Data, Ora
Utente anonimo	Anonimo	Session id	Session id
Chat pubblica	Chatroom globale	Orario scrittura, Ricompense chat, Testo messaggio	Username, Orario scrittura
Chat privata	Conversazione privata	Orario scrittura, Testo messaggio	Username, Orario scrittura, Testo messaggio
Canale	Profilo utente	Follower, Trailer, Descrizione, Social, Immagine profilo	Username
Subscription	Iscrizione	Data inizio, Data fine, Emoji esclusive	Username
Donazione	Contributo monetario	Messaggio, Valore	Username

Dizionario dei dati per le relazioni

Relazione	Descrizione	Componenti	Attributi
Possiede	Possesso di un portafoglio di bit	Utente (1,1), Portafoglio bit (1,1)	N.D.
Pubblica	Pubblicazione di contenuti	Contenuti (1,1), Content creator (0,N)	N.D.
Programma	Invio di un evento nel calendario	Calendario (1,1), Utente (1,1)	N.D.
Guarda	Visualizzazione di una diretta da utente anonimo	Utente anonimo (0,N), Contenuti (0,N)	N.D.
Visualizza	Visualizzazione di una diretta da utente registrato	Contenuti (0,N), Utente (0,N)	N.D.
Riceve	Ricezione di un messaggio pubblico	Live streaming (0,N), Chat pubblica (1,1)	N.D.
Invia	Invio di un messaggio pubblico	Utente (0,N), Chat pubblica (1,1)	N.D.
Scrive	Scrittura di un messaggio privato	Utente (0,N), Chat privata (1,1)	N.D.
Legge	Lettura di un messaggio privato	Chat privata (1,1), Utente (0,N)	N.D.
Gestisce	Gestione e possesso di un canale	Utente (1,1), Canale (1,1)	N.D.
Compra	Acquisto di una subscription o donazione	Donazione (1,1), Subscription (1,1), Utente (0,N)	N.D.
Guadagna	Guadagno di una donazione o di una Subscription	Donazione (1,1), Subscription (1,1), Affiliato (0,N)	N.D.
Segue	Follow di un utente verso un canale	Utente (0,N), Canale (0,N)	N.D.

1.4.3 Vincoli di Integrità

- **Utente:**
 - Deve avere un nome utente univoco.
 - Può seguire altri utenti per ricevere notifiche.
- **Content Creator:**
 - Può guadagnare denaro tramite pubblicità e abbonamenti.
 - Deve seguire le linee guida di Twitch.
- **Affiliato:**
 - Deve rispettare gli accordi di partnership con Twitch.
 - Può guadagnare una percentuale sui ricavi pubblicitari.
 - Deve pubblicare contenuti originali.
- **Contenuti:**
 - I contenuti non devono violare i diritti d'autore.
 - Devono rispettare le linee guida di Twitch.
- **Chat Pubblica:**
 - La chat pubblica non deve contenere linguaggio offensivo o contenuti inappropriati.
- **Chat Privata:**
 - Le chat private non devono essere utilizzate per scopi illegali o dannosi.
- **Subscription:**
 - Le iscrizioni devono essere ottenute onestamente.
- **Donazione:**
 - Non si può effettuare una donazione se il portafoglio non contiene bits.

2 Progettazione Logica

2.1 Tavola dei volumi

Si stima una quantità di risorse partendo da un numero di utenti pari a 5 milioni.

Nota Bene E: Entità R: Relazione

2.2 Tavola dei volumi

Concetto	Tipo	Volume
Utente	E	5.000.000
Content creator	E	500.000
Affiliato	E	400.000
Portafoglio bit	E	5.000.000
Contenuti	E	260.000.000
Video	E	208.000.000
VOD	E	52.000.000
Clip	E	156.000.000
Live streaming	E	52.000.000
Calendario	E	5.000.000
Utente anonimo	E	200.000
Chat pubblica	E	20.000.000.000
Chat privata	E	25.000.000
Canale	E	5.000.000
Subscription	E	4.000.000
Donazione	E	18.750.000
Possiede	R	5.000.000
Pubblica	R	260.000.000
Programma	R	5.000.000
Guarda	R	2.400.000
Visualizza	R	2.880.000.000
Riceve	R	20.000.000.000
Invia	R	20.000.000.000
Scrive	R	25.000.000
Legge	R	25.000.000
Gestisce	R	5.000.000
Compra	R	21.750.000
Guadagna	R	21.750.000
Segue	R	50.000.000

2.3 Spiegazione dei dati

- **Utente:** Per semplificare l'analisi ed utilizzare volumi minori, si suppone che la piattaforma venga utilizzata da 5.000.000 di utenti.
- **Content creator:** Si suppone di avere 500.000 utenti che creano contenuti nella piattaforma.
- **Affiliato:** Suppongo che i content creator affiliati siano circa 400.000.
- **Portafoglio bit:** Ogni utente possiede un portafoglio bit.
- **Contenuti:** Si stima che i contenuti prodotti all'anno da ogni singolo streamer siano 520, ovvero 10 a settimana, di cui 2 dirette, 2 VOD e 6 Clip. Avrò quindi 260.000.000 di contenuti all'anno divisi secondo le stime fatte in precedenza.
- **Video:** Secondo i calcoli fatti in precedenza ho 208.000.000 all'anno.
- **VOD:** Secondo i calcoli fatti in precedenza 52.000.000 ho all'anno.
- **Clip:** Secondo i calcoli fatti in precedenza ho 156.000.000 all'anno.
- **Live streaming:** Secondo i calcoli fatti in precedenza ho 52.000.000 all'anno.
- **Calendario:** Ogni utente possiede un calendario.
- **Utente anonimo:** Si suppone che gli utenti anonimi siano solamente il 5% degli utilizzatori della piattaforma.
- **Chat pubblica:** Si stima che ogni utente invii circa 80 messaggi pubblici a settimana.
- **Chat privata:** Si stima che ogni utente invii solamente 5 messaggi privati l'anno, visto che si tratta di una feature usata rarissimamente.
- **Canale:** Ogni utente possiede un canale.
- **Subscription:** Suppongo che il 50% degli utenti sia disposto a supportare un content creator affiliato. Suppongo che in media ogni utente disposto ha più di un abbonamento all'attivo.
- **Donazione:** Le donazioni tendono ad essere più delle subscription. Suppongo che il 70% degli utenti sia disposto a effettuare una donazione. Suppongo che ogni utente faccia circa 5 donazioni l'anno.
- **Possiede:** Ogni utente possiede uno e un solo portafoglio di bit.
- **Pubblica:** Il numero di pubblicazioni equivale al numero di contenuti presenti nella piattaforma, quindi 260.000.000.
- **Programma:** Ogni utente possiede uno e un solo calendario.
- **Guarda:** Stimo che vista l'assenza di un account, un utente anonimo guardi solamente 1 live al mese in media.
- **Visualizza:** Stimo che ogni utente registrato visualizzi in media 50 contenuti al mese, quindi tutti gli utenti guarderanno 2.880.000.000 contenuti l'anno.
- **Riceve:** Il numero è pari al numero di messaggi presenti nella chat pubblica durante le dirette.
- **Invia:** Il numero è pari al numero di messaggi inviati in media nella chat pubblica durante le dirette.
- **Scrivo:** Il numero è pari al numero di messaggi inviati in media nella chat privata.
- **Legge:** Il numero è pari al numero di messaggi ricevuti in media nella chat privata.
- **Gestisce:** Ogni utente possiede uno e un solo canale.
- **Compra:** Seguendo il punto subscription e donazione spiegato in precedenza, all'anno vengono effettuati $4.000.000 + 18.750.000$ acquisti.
- **Guadagna:** Il numero è lo stesso del punto precedente.
- **Segue:** Stimo che ogni utente segua in media altri 10 canali.

2.4 Tavola delle operazioni

Operazione	Tipo	Frequenza
Controllo delle condizioni per la qualifica di affiliato	I	1/giorno
Calcolo della classifica degli streamer più seguiti	B	1/settimana
Creazione di contenuti	I	712.329/giorno
Visualizzazione dei contenuti	I	7.896.986/giorno
Invio messaggi in chat pubblica	I	54.794.521/giorno
Scrittura messaggi in chat privata	I	68.493/giorno
Follow di un utente verso un canale	I	136.986/giorno

2.5 Ristrutturazione dello schema E-R

2.5.1 Analisi delle ridondanze

Le ridondanze rilevate nel primo schema E-R sono le seguenti:

- L'attributo *Follower* presente nell'entità *Canale* è ridondante poichè il numero di follower guadagnati da un content creator può essere incluso nella relazione *Segue*.
- L'attributo *Spettatori medi* presente nell'entità *Content creator* è ridondante poichè il numero medio di spettatori di uno streamer può essere calcolato attraverso l'attributo *Spettatori simultanei* presente nell'entità *Live streaming*.
- L'attributo *Minuti trasmessi* presente nell'entità *Content creator* è ridondante poichè il numero di minuti trasmessi può essere calcolato attraverso l'attributo *Durata* presente nell'entità *Contenuti*.

Analisi ridondanza *Follower*

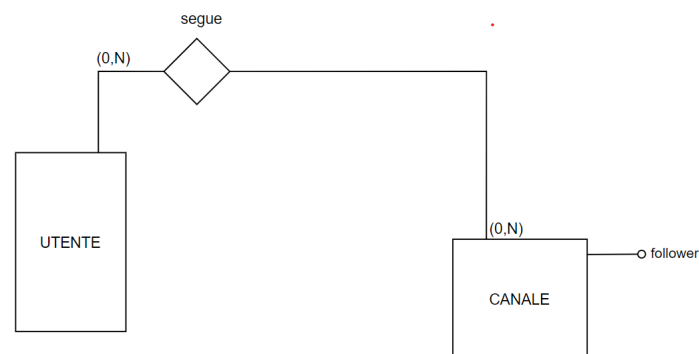


Tavola dei volumi		
Concetto	Tipo	Volume
Utente	E	5.000.000
Canale	E	5.000.000
Segue	R	50.000.000

Tavola delle operazioni		
Operazione	Tipo	Frequenza
Follow di un utente verso un canale	I	136.986/giorno
Calcolo della classifica degli streamer più seguiti	B	1/settimana

Senza ridondanza

Operazione 1: Un utente inizia a seguire un altro utente, in media 136.986 volte al giorno.

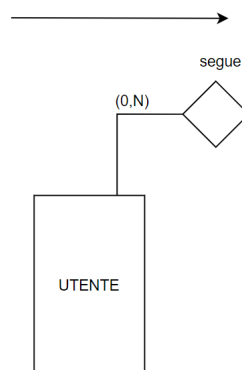


Tavola degli accessi			
Concetto	Costrutto	Accessi	Tipo
Utente	E	1	S
Segue	R	1	S

Nota Bene S: Accesso in scrittura.

Operazione 2: Viene calcolata la classifica dei content creator più seguiti (1 volta a settimana).

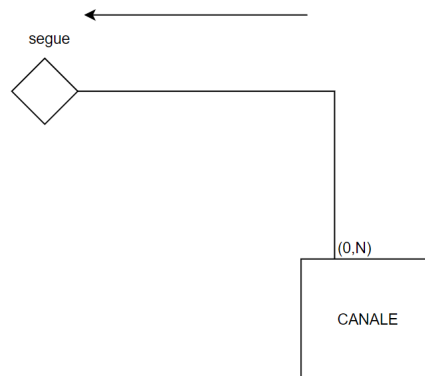


Tavola degli accessi			
Concetto	Costrutto	Accessi	Tipo
Utente	E	1	L
Segue	R	10	L

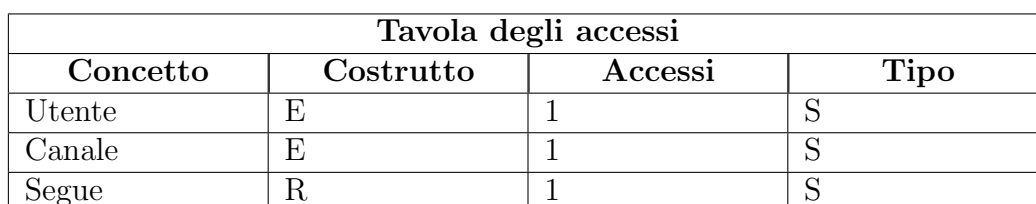
Assumendo che, mediamente, un utente segua 10 canali diversi, gli accessi all'associazione sono dati dal seguente calcolo: $(10 \times Volume.Utente) / Volume.Canale = 10$.

Analisi dei costi:

- Spazio: 0 byte
- Tempo:
 - Operazione 1: $2 \times (2 \times 958.902)$ accessi in lettura a settimana.
 - Operazione 2: 10 accessi in lettura a settimana.

Totale: 3.835.618 accessi a settimana.

Operazione 1: Un utente inizia a seguire un canale, in media 136.986 volte al giorno.



```

graph LR
    CANALE[CANALE] --- follower((follower))
  
```

14

Costi:

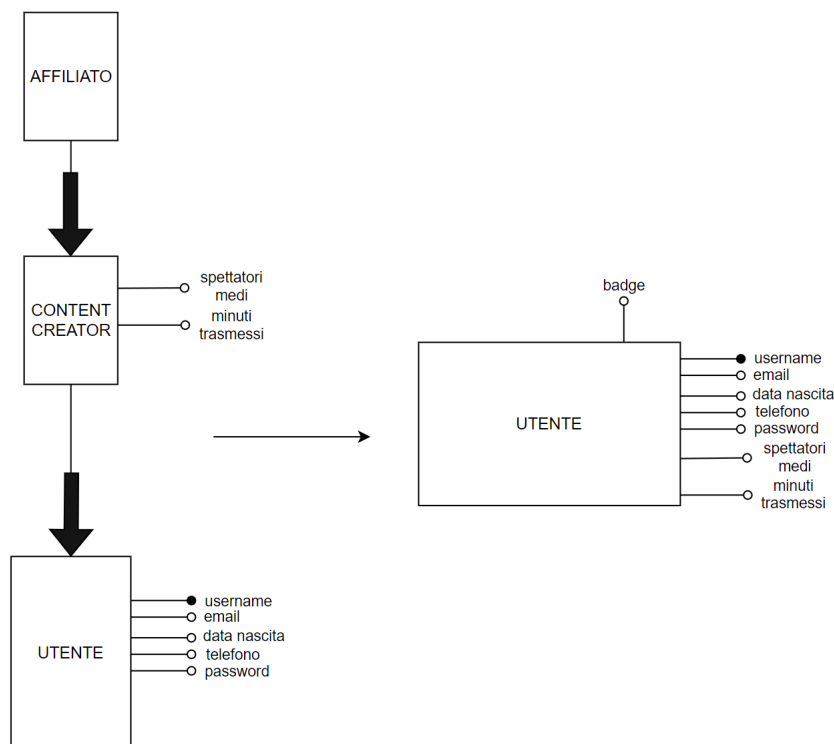
- Spazio: Suppongo di utilizzare 4 byte per il campo *Follower* dell'entità *Canale*, ottengo quindi: $4 \text{ byte} \times 5.000.000 \text{ canali} = 20.000.000 \text{ byte} = 20 \text{ MB}$.
- Tempo:
 - Operazione 1: 3×958.902 accessi in scrittura a settimana.
 - Operazione 2: trascurabile.

Totale: 5.753.412 accessi alla settimana

Conclusione: Eliminando la ridondanza diminuisco significativamente non solo il numero di accessi, ma vado anche a risparmiare 20 MB di spazio.

2.5.2 Eliminazione delle generalizzazioni

Generalizzazione 1:



Business Rules:

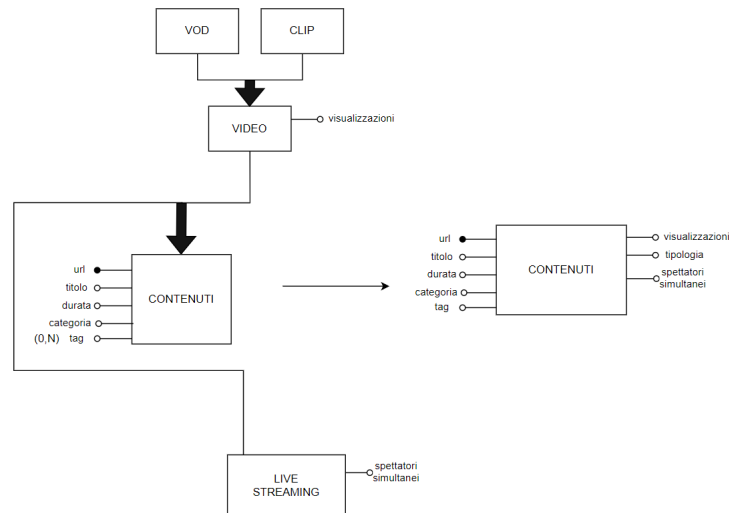
- Un *Utente* può diventare *Affiliato* avendo l'attributo booleano *Affiliato* TRUE.
- Gli attributi restanti vengono ereditati da *Content Creator* ma possono avere valore 0 nel caso in cui *Utente* decida di non creare alcun contenuto.

Motivazione:

L'inclusione delle entità figlie nella loro entità genitore è stata effettuata per ottimizzare lo spazio, mediante la creazione di una nuova tabella che stabilisce una connessione tra

Utente e *Content Creator*. Questo approccio consente l'ereditarietà degli attributi, come *Minuti trasmessi*, *Spettatori medi* e *Affiliato*, da *Utente*. I primi due sono valori interi e possono rimanere a 0 per gli *Utenti* che non generano contenuti. Nel caso in cui un *Utente*, precedentemente *Content creator*, raggiunga gli obiettivi stabiliti, può acquisire lo status di *Affiliato*. La gestione degli *Affiliati* segue un modello simile su tutte le piattaforme online, consentendo loro di guadagnare un semplice *Affiliato*.

Generalizzazione 2:



Business Rules:

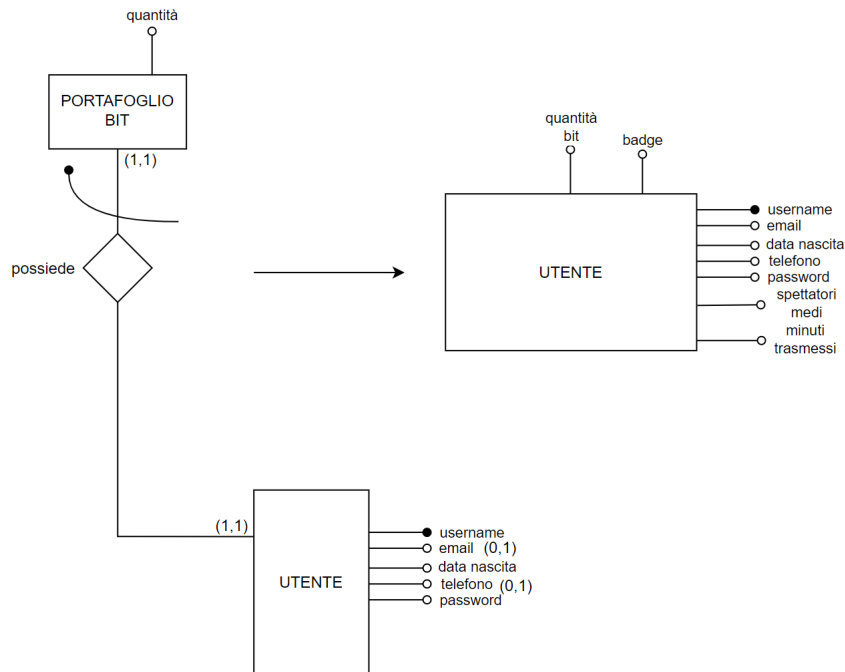
- *Tipologia* è un attributo che uso per descrivere le varie istanze delle entità figlie di *Contenuti*.
- *Spettatori simultanei* viene utilizzato solamente se *Tipologia* ha valore "Live Streaming".
- *Visualizzazioni* viene utilizzato solamente se *Tipologia* ha valore "Video", "Vod" o "Clip".

Motivazione:

Ho optato per un ulteriore raggruppamento delle entità figlie nella loro entità genitore poiché i *Contenuti* presentano attributi comuni a tutti i figli. Inoltre, considerando che le operazioni non discriminano tra le specializzazioni, la decisione più efficiente è risultata essere l'inclusione dei figli nell'entità genitore. Questa scelta ci ha anche portato a capire che effettuare un raggruppamento del genitore nei figli non sarebbe stata la soluzione ottimale.

2.5.3 Eventuale partizionamento/accorpamento di entità e associazioni

Accorpamento 1:



Motivazione:

Ho valutato che le informazioni del *Portafoglio bit* risultano più utili quando sono visualizzate insieme a quelle dell'*Utente*. Considerando il *Portafoglio bit* come un costrutto isolato, offre poche informazioni in quanto richiede sempre lo *Username* per l'accesso. Di conseguenza, poiché ogni *Utente* possiede un unico *Portafoglio bit* e ogni *Portafoglio bit* appartiene a un solo *Utente*, non è necessario creare una nuova tabella per questa entità.

2.5.4 Eventuale eliminazione degli attributi composti e degli attributi multivalore

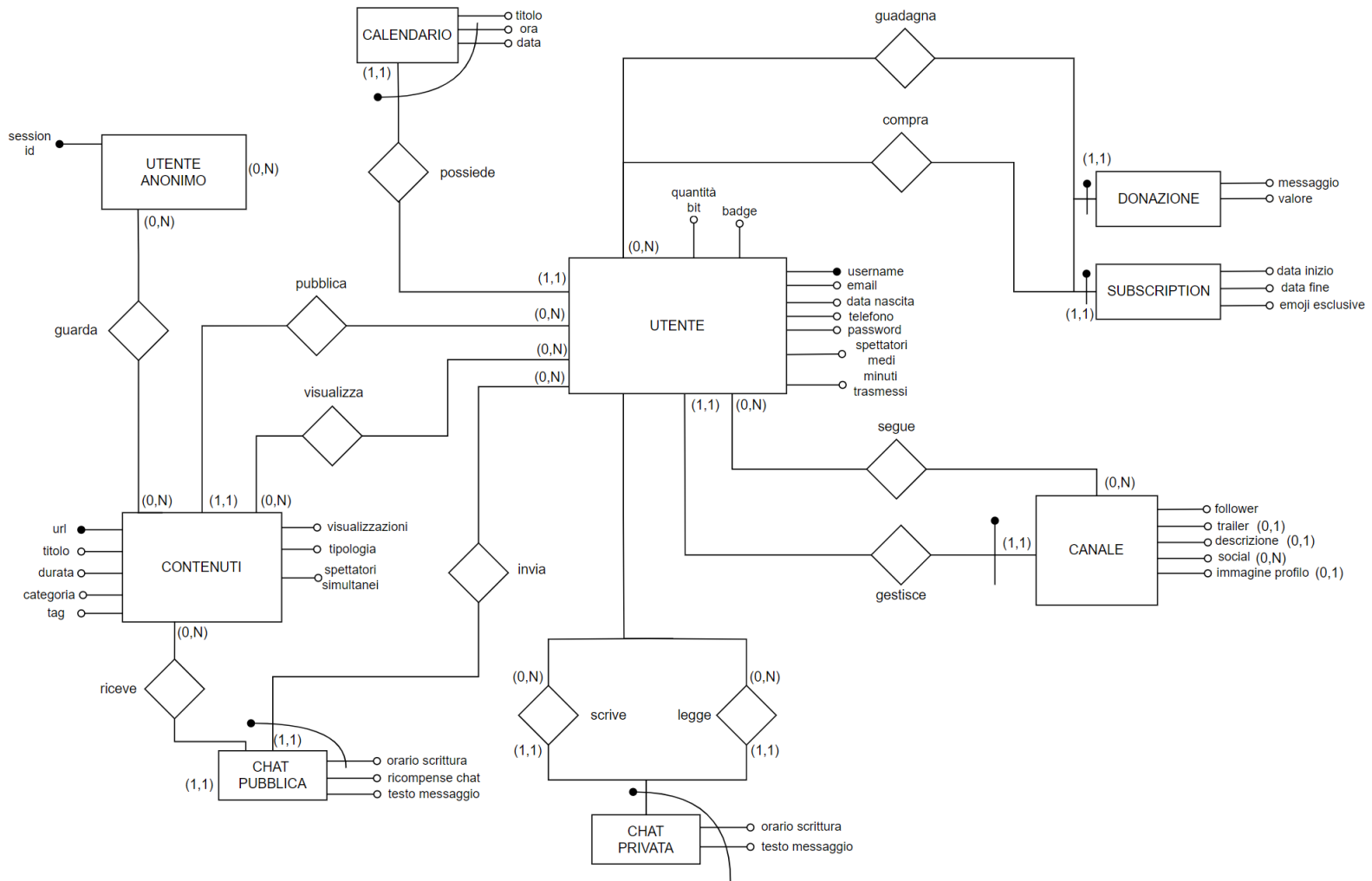
Gli identificatori principali sono quelli indicati come tali nello schema E-R principale.

2.5.5 Eventuale scelta degli identificatori principali

Gli identificatori principali sono quelli indicati come tali nello schema E-R principale.

2.6 Schema E-R ristrutturato + business rules

2.6.1 Schema E-R



2.7 Schema relazionale

- **Utente** (username, email*, telefono*, dataNascita, password, spettatoriMedi, minutiTrasmessi, quantitàBit, affiliato)
- **Calendario** (username, data, ora, titolo)
- **UtenteAnonimo** (sessionId)
- **Contenuti** (url, nomeContentCreator, titolo, durata, categoria, tag, visualizzazioni, tipologia, spettatoriSimultanei)
- **ChatPubblica** (orarioScrittura, utenteMittente, url, ricompenseChat, testoMessaggio)
- **ChatPrivata** (orarioScrittura, testoMessaggio, utenteMittente, utenteDestinatario)
- **Canale** (username, follower, trailer*, descrizione*, immagineProfilo*, social)
- **Subscription** (subber, subbed, dataInizio, dataFine, emojiEsclusive)
- **Donazione** (donatore, beneficiario, messaggio, valore)
- **Guarda** (sessionId, url)
- **Segue** (seguace, seguito)
- **Visualizza** (url, username)

Vincoli di integrità referenziale

- **Calendario** (*username*) referencia **Utente** (*username*)
- **chatPubblica** (*utenteMittente*) referencia **Utente** (*username*)
- **chatPubblica** (*url*) referencia **Contenuti** (*url*)
- **chatPrivata** (*utenteMittente*) referencia **Utente** (*username*)
- **chatPrivata** (*utenteDestinatario*) referencia **Utente** (*username*)
- **Canale** (*username*) referencia **Utente** (*username*)
- **Subscription** (*subber*) referencia **Utente** (*username*)
- **Subscription** (*subbed*) referencia **Utente** (*username*)
- **Donazione** (*donatore*) referencia **Utente** (*username*)
- **Donazione** (*beneficiario*) referencia **Utente** (*username*)
- **Guarda** (*sessionId*) referencia **UtenteAnonimo** (*sessionId*)
- **Guarda** (*url*) referencia **Contenuti** (*url*)
- **Segue** (*seguace*) referencia **Utente** (*username*)
- **Segue** (*seguito*) referencia **Utente** (*username*)
- **Visualizza** (*url*) referencia **Contenuti** (*url*)
- **Visualizza** (*username*) referencia **Utente** (*username*)

3 Implementazione

3.1 DDL di creazione del database

```
DROP TABLE IF EXISTS utente CASCADE;
DROP TABLE IF EXISTS utenteAnonimo CASCADE;
DROP TABLE IF EXISTS canale CASCADE;
DROP TABLE IF EXISTS calendario CASCADE;
DROP TABLE IF EXISTS contenuti CASCADE;
DROP TABLE IF EXISTS chatPubblica CASCADE;
DROP TABLE IF EXISTS chatPrivata CASCADE;
DROP TABLE IF EXISTS subscription CASCADE;
DROP TABLE IF EXISTS donazione CASCADE;
DROP TABLE IF EXISTS guarda CASCADE;
DROP TABLE IF EXISTS pubblica CASCADE;
DROP TABLE IF EXISTS segue CASCADE;
DROP TABLE IF EXISTS visualizza CASCADE;
```

```
CREATE TABLE utente(
    username varchar PRIMARY KEY,
    email varchar DEFAULT NULL,
    telefono varchar DEFAULT NULL,
    dataNascita date NOT NULL,
    password varchar NOT NULL,
    spettatoriMedi integer DEFAULT 0,
    minutiTrasmessi integer DEFAULT 0,
    quantitaBit integer DEFAULT 0,
    affiliato boolean DEFAULT false
);
```

```
CREATE TABLE calendario(
    username varchar,
    data date,
    ora time,
    titolo varchar NOT NULL,
    PRIMARY KEY (username, data, ora)
);
```

```
CREATE TABLE utenteAnonimo(
    sessionId varchar PRIMARY KEY
);
```

```
CREATE TABLE contenuti(
    url varchar PRIMARY KEY,
    nomeContentCreator varchar NOT NULL,
    titolo varchar NOT NULL,
    durata time NOT NULL,
    categoria varchar NOT NULL,
```

```

    tag varchar NOT NULL,
    visualizzazioni integer DEFAULT 0,
    tipologia varchar NOT NULL,
    spettatoriSimultanei integer DEFAULT 0
);

CREATE TABLE chatPubblica(
    orarioScrittura time,
    utenteMittente varchar,
    url varchar,
    ricompenseChat varchar NOT NULL,
    testoMessaggio varchar NOT NULL,
    PRIMARY KEY (orarioScrittura, utenteMittente, url)
);

CREATE TABLE chatPrivata(
    orarioScrittura time,
    testoMessaggio varchar,
    utenteMittente varchar,
    utenteDestinatario varchar,
    PRIMARY KEY (orarioScrittura, testoMessaggio,
    utenteMittente, utenteDestinatario)
);

CREATE TABLE canale(
    username varchar PRIMARY KEY,
    follower integer DEFAULT 0,
    trailer varchar DEFAULT NULL,
    immagineProfilo varchar DEFAULT NULL,
    social varchar DEFAULT NULL,
    descrizione varchar DEFAULT NULL
);

CREATE TABLE subscription(
    subber varchar,
    subbed varchar,
    dataInizio date DEFAULT NULL,
    dataFine date DEFAULT NULL,
    emojiEsclusive varchar DEFAULT NULL,
    PRIMARY KEY (subber, subbed)
);

CREATE TABLE donazione(
    donatore varchar,
    beneficiario varchar,
    messaggio varchar DEFAULT NULL,
    valore integer DEFAULT 0,
    PRIMARY KEY (donatore, beneficiario)
);

```

```
);
```

```
CREATE TABLE guarda(  
    sessionID varchar,  
    url varchar,  
    PRIMARY KEY (sessionID, url)  
);
```

```
CREATE TABLE segue(  
    seguace varchar,  
    seguito varchar,  
    PRIMARY KEY (seguace, seguito)  
);
```

```
CREATE TABLE visualizza(  
    url varchar,  
    username varchar,  
    PRIMARY KEY (url, username)  
);
```

```
ALTER TABLE calendario  
ADD FOREIGN KEY (username) REFERENCES utente (username)  
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE chatPubblica  
ADD FOREIGN KEY (utenteMittente) REFERENCES utente (username)  
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE chatPrivata  
ADD FOREIGN KEY (utenteMittente) REFERENCES utente (username)  
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE chatPrivata  
ADD FOREIGN KEY (utenteDestinatario) REFERENCES utente (username)  
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE canale  
ADD FOREIGN KEY (username) REFERENCES utente (username)  
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE subscription  
ADD FOREIGN KEY (subber) REFERENCES utente (username)  
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE subscription  
ADD FOREIGN KEY (subbed) REFERENCES utente (username)
```

```

ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE donazione
ADD FOREIGN KEY (donatore) REFERENCES utente (username)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE donazione
ADD FOREIGN KEY (beneficiario) REFERENCES utente (username)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE guarda
ADD FOREIGN KEY (sessionId) REFERENCES utenteAnonimo (sessionId)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE guarda
ADD FOREIGN KEY (url) REFERENCES contenuti (url)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE segue
ADD FOREIGN KEY (seguace) REFERENCES utente (username)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE segue
ADD FOREIGN KEY (seguito) REFERENCES utente (username)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE visualizza
ADD FOREIGN KEY (url) REFERENCES contenuti (url)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE visualizza
ADD FOREIGN KEY (username) REFERENCES utente (username)
ON DELETE CASCADE ON UPDATE CASCADE;

```

3.2 DML di popolamento di tutte le tabelle del database

```

INSERT INTO utente VALUES ('oskar.heise', 'oskar@gmail.com',
'1234567890', '2002-12-18', 'P4ssw0rd??', 0, 0, 50, false);
INSERT INTO utente VALUES ('mario.rossi', 'mario.rossi@gmail.com',
'9876543210', '1995-05-22', 'M@rioP@ss!', 0, 0, 50, false);
INSERT INTO utente VALUES ('laura.bianchi', 'lauri@yahoo.com',
'7654321098', '1988-11-10', 'L@uraP@ss#', 0, 0, 50, false);
INSERT INTO utente VALUES ('giuseppe.verdi', 'gius.verdi@hotmail.com',
'8765432109', '1980-03-15', 'Giuseppe#123', 0, 0, 50, false);
INSERT INTO utente VALUES ('francesca.rossini', 'fra.ross@gmail.com',
'6543210987', '1992-07-03', 'Fr@ncescaPwd', 0, 0, 50, false);
INSERT INTO utente VALUES ('marco.gallo', 'marco.gallo@gmail.com',

```



```

'7890123456', '1985-09-28', 'M@rcoGallo!', 0, 0, 50, false);
INSERT INTO utente VALUES ('anna.ferrari', 'anna.ferrari@yahoo.com',
'8901234567', '1990-01-12', 'AnnaF3rr@ri', 0, 0, 50, false);
INSERT INTO utente VALUES ('luca.martinelli', 'luca.ma@hotmail.com',
'5678901234', '1987-04-08', 'Luca_M@rtin', 0, 0, 50, false);
INSERT INTO utente VALUES ('simona.ricci', 'simona.ricci@gmail.com',
'6789012345', '1983-06-20', 'Sim0naRicci*', 0, 0, 50, false);
INSERT INTO utente VALUES ('paolo.colombo', 'paolo.colo@yahoo.com',
'3456789012', '1998-02-25', 'Pa0loC0lomb0#', 0, 0, 50, false);
INSERT INTO utente VALUES ('roberta.mazza', 'rob.mazza@hotmail.com',
'4567890123', '1982-10-07', 'R0bertaM@zz@', 0, 0, 50, false);
INSERT INTO utente VALUES ('andrea.romano', 'and.roma@gmail.com',
'2345678901', '1993-12-30', 'Andr34R0mano$', 0, 0, 50, false);
INSERT INTO utente VALUES ('elisa.rizzo', 'eli.rizzo@yahoo.com',
'4321098765', '1989-08-18', '3lisaRizz0*', 0, 0, 50, false);
INSERT INTO utente VALUES ('giorgio.conti', 'gio.con@hotmail.com',
'3210987654', '1981-04-05', 'Gi0rgioC0nti!', 0, 0, 50, false);
INSERT INTO utente VALUES ('veronica.santoro', 'vero.sant@gmail.com',
'2109876543', '1996-06-15', 'V3ron!caS@nt0r0', 0, 0, 50, false);
INSERT INTO utente VALUES ('davide.ferri', 'da.fer@yahoo.com',
'1098765432', '1984-09-12', 'D@videF3rri#', 0, 0, 50, false);
INSERT INTO utente VALUES ('sara.lombardi', 'sara.lomba@hotmail.com',
'9012345678', '1991-11-28', 'S@raL0mbardi$', 0, 0, 50, false);
INSERT INTO utente VALUES ('marcello.gatti', 'marce@gmail.com',
'5432109876', '1986-02-14', 'M@rc3ll0G@tti', 0, 0, 50, false);
INSERT INTO utente VALUES ('giulia.palumbo', 'giu.palu@yahoo.com',
'6789012345', '1997-03-08', 'Giuli@P@lumb0', 0, 0, 50, false);
INSERT INTO utente VALUES ('antonio.rocce', 'anto.rocce@hotmail.com',
'1234567890', '1980-07-23', 'Ant0ni0R0cca*', 0, 0, 50, false);
INSERT INTO utente VALUES ('eleonora.damico', 'ele.dami@gmail.com',
'2345678901', '1994-05-18', 'El30n0raD@mico#', 0, 0, 50, false);
INSERT INTO utente VALUES ('marco.ferretti', 'marco.ferre@yahoo.com',
'9876543210', '1987-08-15', 'M@rcoFerr3tti#', 0, 0, 50, false);
INSERT INTO utente VALUES ('silvia.monti', 'silviam@hotmail.com',
'8765432109', '1995-02-02', 'S!lviaM0nti*', 0, 0, 50, false);
INSERT INTO utente VALUES ('andrea.palazzi', 'andr.pala@gmail.com',
'7654321098', '1983-11-20', 'Andr3aP@l@zzi!', 0, 0, 50, false);
INSERT INTO utente VALUES ('valentina.battaglia', 'valenb@yahoo.com',
'6543210987', '1990-04-12', 'V@l3ntinaB#tt@glia', 0, 0, 50, false);
INSERT INTO utente VALUES ('gianni.moretti', 'giannim@gmail.com',
'8901234567', '1982-06-25', 'Gi@nniM0rett!$', 0, 0, 50, false);
INSERT INTO utente VALUES ('francesca.ferri', 'fra.fe@hotmail.com',
'5678901234', '1991-09-18', 'Fr@nc3scaF3rri#', 0, 0, 50, false);
INSERT INTO utente VALUES ('luca.ricci', 'luca.ricci@yahoo.com',
'6789012345', '1988-03-03', 'L#caR1cci*', 0, 0, 50, false);
INSERT INTO utente VALUES ('anna.martini', 'ann.marti@gmail.com',
'3456789012', '1996-07-22', 'AnnaM@rtini123', 0, 0, 50, false);
INSERT INTO utente VALUES ('marco.rossi', 'marco.rossi@hotmail.com',

```

```

'4567890123', '1985-12-10', 'M@rcoR0ss!123', 0, 0, 50, false);
INSERT INTO utente VALUES ('elena.galli', 'ele.gal@yahoo.com',
'2345678901', '1993-01-28', '3lenaG@lli#', 0, 0, 50, false);

INSERT INTO utente VALUES ('ninja', 'ninja@gmail.com', '9876543210',
'1991-06-05', 'N!nj@P@ss123', 2000, 1000, 500, true);
INSERT INTO utente VALUES ('shroud', 'shroud@yahoo.com', '8765432109',
'1994-11-02', 'Shr0ud#987', 1500, 1000, 250, true);
INSERT INTO utente VALUES ('tfue', 'tfue@hotmail.com', '7654321098',
'1997-01-02', 'TfueP@ss!321', 1800, 1200, 400, true);
INSERT INTO utente VALUES ('xqcow', 'xqcow@hotmail.com', '5678901234',
'1996-11-12', 'XqC0wP@ss#', 2500, 1000, 650, true);
INSERT INTO utente VALUES ('lirik', 'lirik@gmail.com', '6789012345',
'1990-10-29', 'L!r!kP@ss456', 800, 1200, 100, false);

INSERT INTO calendario VALUES ('ninja', '2024-11-14',
'20:00:00', 'NuovaLive');

INSERT INTO utenteAnonimo VALUES ('ajkdsfh7819');
INSERT INTO utenteAnonimo VALUES ('jasdksd1617');

INSERT INTO contenuti VALUES ('www.twitch.tv/videos/1887789454',
'ninja', 'TalkShow', '02:00:12', 'JustChatting',
'QuattroChiacchiere', 0, 'Live_streaming', 1200);
INSERT INTO contenuti VALUES ('www.twitch.tv/videos/1844449454',
'tfue', 'Gaming', '02:00:12', 'Gaming', 'Giochiamo',
1500, 'VOD', 0);
INSERT INTO contenuti VALUES ('www.twitch.tv/videos/1823349454',
'tfue', 'Gaming', '00:00:15', 'Gaming', 'Giochiamo',
300, 'clip', 0);

INSERT INTO chatPubblica VALUES ('00:30:23', 'oskar.heise',
'www.twitch.tv/videos/1887789454', 0, 'Buonasera');

INSERT INTO chatPrivata VALUES ('12:30:00', 'ciao!', 'oskar.heise',
'davide.ferri');

INSERT INTO canale VALUES ('oskar.heise');
INSERT INTO canale VALUES ('mario.rossi');
INSERT INTO canale VALUES ('laura.bianchi');
INSERT INTO canale VALUES ('giuseppe.verdi');
INSERT INTO canale VALUES ('francesca.rossini');
INSERT INTO canale VALUES ('marco.gallo');
INSERT INTO canale VALUES ('anna.ferrari');
INSERT INTO canale VALUES ('luca.martinelli');
INSERT INTO canale VALUES ('simona.ricci');

```

```

INSERT INTO canale VALUES ('paolo.colombo');
INSERT INTO canale VALUES ('roberta.mazza');
INSERT INTO canale VALUES ('andrea.romano');
INSERT INTO canale VALUES ('elisa.rizzo');
INSERT INTO canale VALUES ('giorgio.conti');
INSERT INTO canale VALUES ('veronica.santoro');
INSERT INTO canale VALUES ('davide.ferri');
INSERT INTO canale VALUES ('sara.lombardi');
INSERT INTO canale VALUES ('marcello.gatti');
INSERT INTO canale VALUES ('giulia.palumbo');
INSERT INTO canale VALUES ('antonio.rocce');
INSERT INTO canale VALUES ('eleonora.damico');
INSERT INTO canale VALUES ('marco.ferretti');
INSERT INTO canale VALUES ('silvia.monti');
INSERT INTO canale VALUES ('andrea.palazzi');
INSERT INTO canale VALUES ('valentina.battaglia');
INSERT INTO canale VALUES ('gianni.moretti');
INSERT INTO canale VALUES ('francesca.ferri');
INSERT INTO canale VALUES ('luca.ricci');
INSERT INTO canale VALUES ('anna.martini');
INSERT INTO canale VALUES ('marco.rossi');
INSERT INTO canale VALUES ('elena.galli');
INSERT INTO canale VALUES ('ninja');
INSERT INTO canale VALUES ('shroud');
INSERT INTO canale VALUES ('tfue');
INSERT INTO canale VALUES ('xqcow');
INSERT INTO canale VALUES ('lirik');

INSERT INTO subscription VALUES ('oskar.heise', 'ninja',
CURRENT_TIMESTAMP, CURRENT_TIMESTAMP + INTERVAL '1 month', 4.99);

INSERT INTO donazione VALUES ('oskar.heise',
'tfue', 'Evvai', 9.99);

INSERT INTO guarda VALUES ('ajkdsfh7819',
'www.twitch.tv/videos/1887789454');

INSERT INTO visualizza VALUES ('www.twitch.tv/videos/1887789454',
'oskar.heise');

INSERT INTO segue VALUES ('oskar.heise',
'ninja');
INSERT INTO segue VALUES ('oskar.heise',
'shroud');
INSERT INTO segue VALUES ('oskar.heise',
'tfue');
INSERT INTO segue VALUES ('oskar.heise',

```

```

'likrik');
INSERT INTO segue VALUES ('oskar.heise',
'xqcow');
INSERT INTO segue VALUES ('oskar.heise',
'francesca.ferri');
INSERT INTO segue VALUES ('oskar.heise',
'eleonora.damico');
INSERT INTO segue VALUES ('oskar.heise',
'andrea.palazzi');
INSERT INTO segue VALUES ('oskar.heise',
'giulia.palumbo');
INSERT INTO segue VALUES ('oskar.heise',
'simona.ricci');

```

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

```

UPDATE utente
SET username = 'ninjaTwitch'
WHERE username = 'ninja';

```

```

UPDATE utente
SET username = 'OskarConLaK'
WHERE username = 'oskar.heise';

```

```

UPDATE utente
SET quantitaBit = 100
WHERE username = 'OskarConLaK';

```

4 Extra

Classifica dei content creator più seguiti

```

SELECT DENSE_RANK() OVER(ORDER BY COUNT(*) DESC) AS Classifica,
u.username AS username, COUNT(*) as follower
FROM utente u JOIN follower f ON f.follower = s.username
GROUP BY s.username
LIMIT 25;

```