

Relazione del progetto di laboratorio Base di Dati

a.a 22/23

Abourida Zakaria

950120

Progettazione Concettuale

1.1) Requisiti iniziali

Laboratorio Basi di Dati 2022/2023

Progetto di piattaforma di live streaming

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 (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.

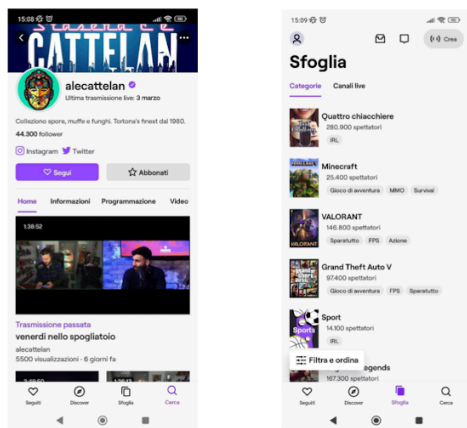


Figura 1 (a) La home del canale di aleccattalan. (b) La lista delle categorie

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

"STREAMER"

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 affiliato. 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 affiliato.
- 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	Persona che si registra alla piattaforma che ha diritto a premi...	CLIENTE, ABBONATO, CONSUMATORE	SPETTATORE STREAMER VIEWER CANALE CONTENUTI
STREAMER	Persona (utente) che pubblica i suoi contenuti sulla piattaforma		UTENTE

TERMINE	DESCRIZIONE	SINONIMI	COLLEGAMENTI
VIEWER	Persona (utente) che interagisce e guarda i contenuti online	TELESPETTATORE	UTENTE
SPETTATORE	Persona (non utente) che può solo guardare i contenuti senza interagire	OSSERVATORE	CONTENUTI UTENTE
CANALE	Profilo dell'utente in cui carica le sue live ed i suoi video	PROFILO	UTENTE CONTENUTI
CONTENUTI	Pubblicazioni degli utenti nella piattaforma	ARGOMENTI	CANALE UTENTE SPETTATORE LIVE VIDEO
LIVE	Video in diretta da parte degli utenti	DIRETTA, TRASMISSIONE	CONTENUTI
VIDEO	Live trasformata in video al suo termine per la conservazione	CINEMATOGRAFIA	CONTENUTI

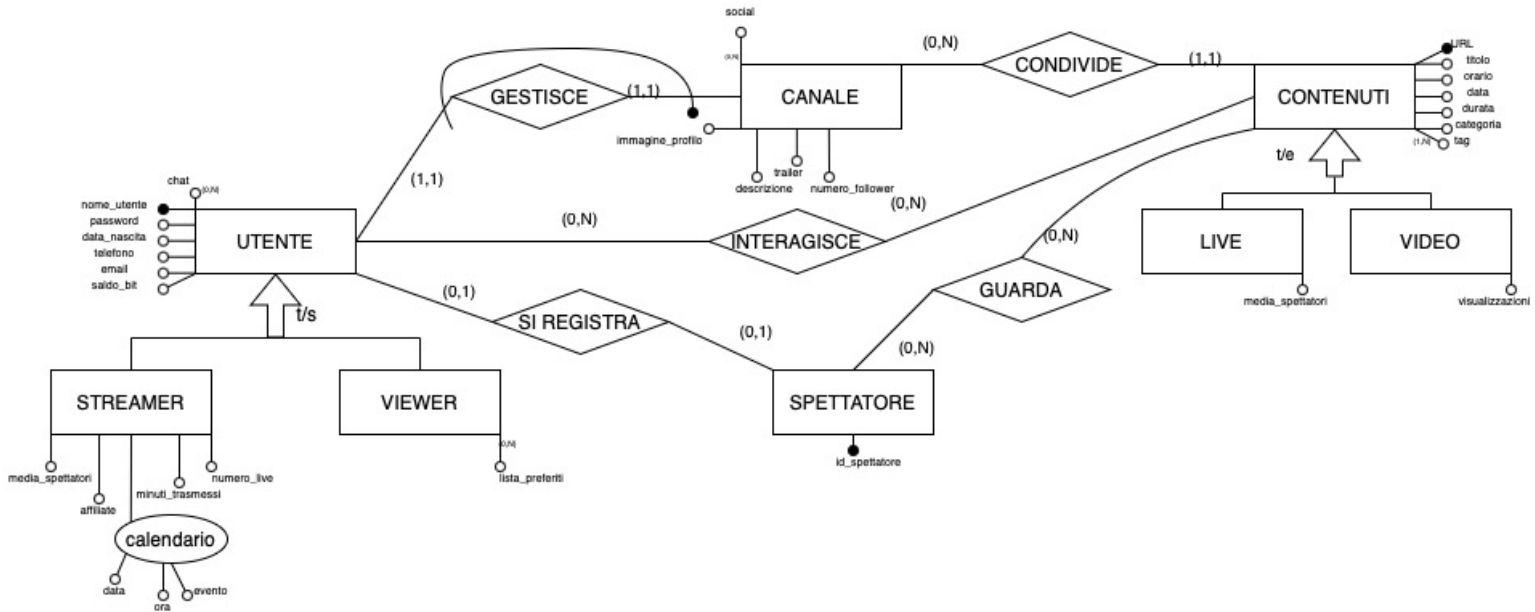
1.3) Requisiti rivisti e strutturati in gruppi di frasi omogenee

- Frasi per UTENTE:
 1. Ogni utente può essere STREAMER o VIEWER, o entrambi
 2. Devono indicare nome utente, password, data di nascita, telefono o email
 3. Possono chattare pubblicamente o privatamente tra loro
 4. Possono seguire lo STREAMER (subscription)
 5. Possono creare dirette
 6. Hanno un portafoglio di bit per poterli donare agli STREAMER
- Frasi per STREAMER:
 1. Hanno un CANALE
 2. Viene rappresentato il numero di live fatte
 3. Viene rappresentato il numero di minuti trasmessi
 4. Viene rappresentato il numero medio di spettatori simultanei

5. Se ha minimo 500 minuti trasmessi, una media superiore/
uguale a 3 spettatori simultanei ed un minimo di 50 follower allora
può diventare affiliate
 6. Possiede un calendario
- Frasi per VIEWER:
 1. Può diventare follower degli streamer (elenco di follower /
preferiti)
 2. Possono supportare gli STREAMER facendo subscription per
avere privilegi, sticker...
 - Frasi per SPETTATORE:
 1. Può registrarsi e diventare un utente o altrimenti guardare i
CONTENUTI in anonimo
 - Frasi per CANALE:
 1. Possiede una descrizione
 2. Possiede una lista di social
 3. Possiede un'immagine profilo
 4. Possiede un trailer
 5. Possiede un indicatore per il numero di follower
 - Frasi per LIVE:
 1. Al termine possono diventare VIDEO o no
 2. Possiede un titolo
 3. Possiede una durata
 4. Possiede una categoria
 5. Possiede una lista di tag
 6. Possiede un indicatore per il numero medio di spettatori
 7. Possiede la data e l'ora
 - Frasi per VIDEO:
 1. Possiede un titolo
 2. Possiede una durata

3. Possiede una categoria
4. Possiede una lista di tag
5. Possiede un indicatore per il numero di visualizzazioni

1.4) Schema E/R principale + Business Rule



BUSINESS RULE:

- Per registrarsi gli utenti devono inserire tutti i dati e scegliere se mettere il telefono e/o l'email
- Quando uno streamer raggiunge almeno 500 minuti trasmessi, una media superiore/uguale a 3 spettatori simultanei ed un minimo di 50 follower allora può diventare affiliato
- I viewer ottengono privilegi se effettuano la subscription
- Gli utenti possono seguire lo streamer e creare dirette
- Le live possono diventare o no dei video
- Un video di breve durata è una clip (<30sec)
- Gli utenti non possono donare bit se il loro portafoglio è vuoto o il saldo non è sufficiente

- Gli streamer possono programmare le live aggiungendo gli eventi sul calendario
- Uno spettatore può registrarsi alla piattaforma e diventare un utente, ma un utente non può essere uno spettatore.

REGOLE DI DERIVAZIONE:

- Per ogni live viene calcolato il numero medio di spettatori
- Per ogni video viene calcolato il numero di visualizzazioni
- Per gli streamer vengono calcolati il numero di live effettuate, il numero di minuti trasmessi ed il numero medio di spettatori simultanei

Progettazione Logica

2.1) Tavola dei volumi

CONCETTO	TIPO	VOLUME	DESCRIZIONE
UTENTE	E	15'000'000	Stima degli utenti registrati
STREAMER	E	10'500'000	Stima degli streamer attivi
VIEWER	E	15'000'000	Stima dei viewer (attivi o meno)
CANALE	E	15'000'000	Ogni utente ha un canale (che lo utilizzi o meno)
SPETTATORE	E	7'000'000	Stima degli spettatori anonimi
CONTENUTI	E	1'000'000	Stima dei contenuti pubblicati al giorno tra live e video
LIVE	E	750'000	All'incirca 2/3 dei contenuti sono live
VIDEO	E	250'000	All'incirca 1/3 dei contenuti sono video/clip

CONCETTO	TIPO	VOLUME	DESCRIZIONE
GESTISCE	A	15'000'000	Per ogni utente un canale
INTERAGISCE	A	16'000'000	Stima di almeno 1/2 interazioni per viewer attivo
SI_REGISTRA	A	3'000	Stima delle registrazioni settimanali (spettatore -> utente)
CONDIVIDE	A	750'000	Stima delle condivisioni da parte degli streamer giornaliere
GUARDA	A	12'000'000	Per ogni spettatore guarda almeno 1/2 live

2.2) Tavola delle operazioni

OPERAZIONE	DESCRIZIONE	TIPO	FREQUENZA
1	Registrazione utente	I	1000/settimana
2	Accesso utente	I	2000/settimana
3	Modifica profilo (descrizione, trailer...)	I	2000/settimana
4	Modifica impostazioni	I	1500/settimana
5	Elimina utente	I	500/anno
6	Modifica canale	I	40'000/giorno
7	Aggiorna follower	I	1/ora
8	Crea live	I	5000/settimana
9	Crea video/clip	I	4000/settimana
10	Modifica live	I	5000/settimana
11	Modifica video/clip	I	5000/settimana
12	Elimina live	I	100/mese
13	Elimina video/clip	I	100/mese

OPERAZIONE	DESCRIZIONE	TIPO	FREQUENZA
14	Rendi live in video	I	300/settimana
15	Messaggio in chat pubblica	I	3000/giorno
16	Messaggi privati	I	400/giorno
17	Fai follow	I	100/settimana
18	Subscription	I	500/settimana
19	Donazione	I	5000/settimana
20	Aggiorna visualizzazioni	B	1/ora
21	Controllo affiliate	B	1/giorno
22	Calcolo classifica streamer	B	1/settimana
23	Calcolo numero live effettuate	B	1/giorno
24	Calcolo media spettatori simultanei	B	1/ora
25	Calcolo media spettatori simultanei	B	1/giorno
26	Calcolo numero visualizzazioni	B	1/giorno
27	Aggiorna spettatori in live	B	1/ora
28	Calcolo numero minuti trasmessi	B	1/gg

***La frequenza è stata stimata basandosi sui numeri presenti nella tavola dei volumi e preso spunto da vari siti di report su internet.

Fonti:

- <https://www.businessofapps.com/data/twitch-statistics/>
- <https://twitchtracker.com/statistics/viewers>
- Altre fonti come immagini, notizie...

2.3.1) Analisi delle ridondanze

- STREAMER (numero_live)
- STREAMER (minuti_trasmessi)
- STREAMER(media_spettatori)

Analisi della prima ridondanza->

Tavola dei volumi d'interesse:

CONCETTO	TIPO	VOLUME
STREAMER	E	15'000'000
UTENTE	E	15'000'000
GESTISCE	A	15'000'000
CANALE	E	15'000'000
CONDIVIDE	A	450'000
CONTENUTI	E	450'000
LIVE	E	300'000

Tavola delle operazioni d'interesse:

OPERAZIONE	DESCRIZIONE	TIPO	FREQUENZA
8	Crea live	I	300'000/giorno
12	Elimina live	I	5'000/settimana
23	Calcola numero live effettuate	B	1/giorno

Scenario A: senza ridondanza

- Tavola accessi:
- Operazione n'8:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
LIVE	E	1	S
CONTENUTI	E	1	S

CONCETTO	COSTRUTTO	ACCESSI	TIPO
CONDIVIDE	A	1	S
CANALE	E	1	L
GESTISCE	A	1	L
UTENTE	E	1	L
STREAMER	E	1	L
STREAMER	E	1	S

- Operazione n'12:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
LIVE	E	1	S
CONTENUTI	E	1	S
CONDIVIDE	A	1	S
CANALE	E	1	L
GESTISCE	A	1	L
UTENTE	E	1	L
STREAMER	E	1	L
STREAMER	E	1	S

- Operazione n'23:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
STREAMER	E	1	L

Scenario B: con ridondanza

- Tavola accessi:

- Operazione n'8:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
LIVE	E	1	S
CONTENUTI	E	1	S
CONDIVIDE	A	1	S

- Operazione n'12:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
LIVE	E	1	S
CONTENUTI	E	1	S
CONDIVIDE	A	1	S

- Operazione n'23:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
CANALER	E	1	L
CONDIVIDE	A	33	L
CONTENUTI	E	1	L
LIVE	E	1	L

***333 = 15000000 / 450000 (V_canale / V_contenuti)

- Costi:

***Gli accessi in scrittura sono più onerosi quindi per fare una stima dei costi contiamo il costo di una scrittura come il doppio di una lettura

A) - Spazio: 0 byte perché non essendoci l'attributo non occupa spazio

- Tempo:

- op. 8: $3 \cdot 300000$ (3 scritture * 300000 accessi/giorno)
- op. 12: $3 \cdot 714$ (3 scritture * 714 accessi/giorno (= 5000/settimana))
- op. 23: $4 \cdot 36$ (4 letture * (33 + 3 accessi/giorno)

-> in totale ci sono $(900000+2142) \cdot 2 + 144 = 1'804'428$ accessi al giorno

B) - Spazio: contando all'incirca 4 byte per memorizzare il numero di live effettuate per ogni streamer si conta $4 \cdot 15000000 = 60'000'000$ byte necessari (0.06 GB)

- Tempo:

- op. 8: $4 \cdot 300000 + 4 \cdot 300000$
- op. 12: $4 \cdot 714 + 4 \cdot 714$
- op. 23: trascurabile

-> in totale ci sono $(1200000+2856)*2 + (1200000+2856) = 3'608'568$ accessi al giorno

Possiamo concludere che tenere la ridondanza implica una maggiore quantità di spazio occupato ed un tempo di accesso maggiore (quasi il doppio) rispetto al mantenere aggiornato il dato ridondante tramite il ricalcolo di esso.

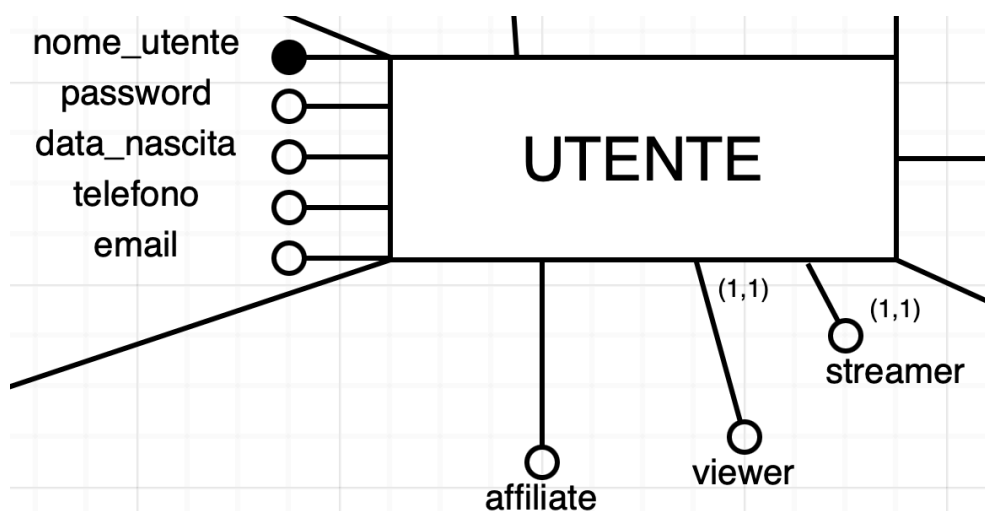
Dunque ho deciso di rimuovere la prima ridondanza per cercare di mantenere una base di dati ottimizzata e veloce.

Per quanto riguarda la seconda e la terza ridondanza, l'analisi è abbastanza analoga a quella della prima ridondanza, dunque ho deciso di rimuoverle per le stesse motivazioni della prima.

2.3.2) Eliminazione delle generalizzazioni

1. UTENTE → STREAMER / VIEWER (totale / sovrapposta)
2. CONTENUTI → LIVE / VIDEO (totale / esclusiva)

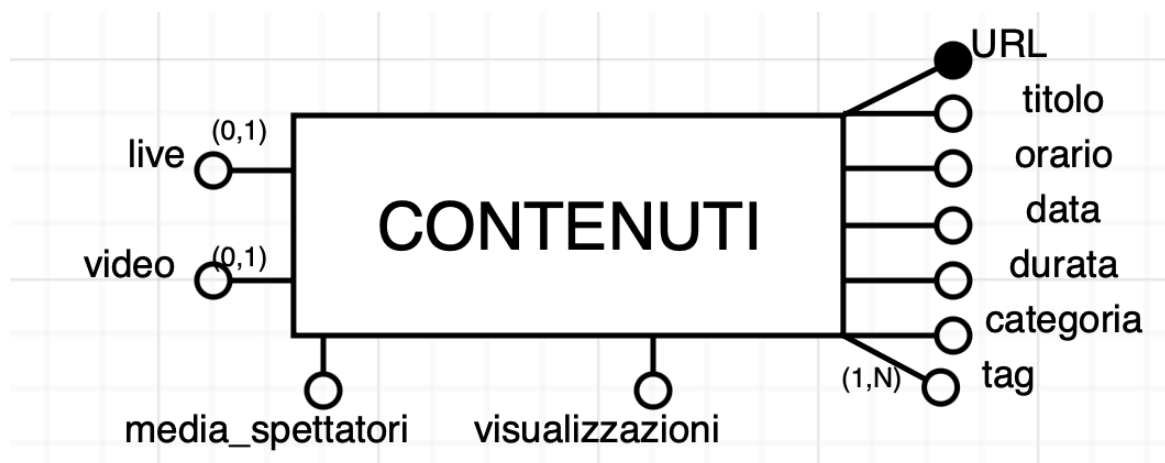
1) Per quanto riguarda la prima generalizzazione ho deciso di utilizzare l'accorpamento dei figli nel padre. Ciò conviene perché non si fa distinzione tra le operazioni. Un utente è sia streamer che viewer nel momento in cui s'iscrive. Che poi sia uno streamer/viewer attivo o meno non ci interessa perché si stima un "piccolo" spreco di spazio per i dati nulli. In questo modo le transazioni possono accedere a tutti i dati in un'unica tabella relazionale riducendo gli accessi.



Aggiunte Business Rule:

- L'attributo streamer ha cardinalità (1,1)
- L'attributo viewer ha cardinalità (1,1)
- Gli attributi streamer e viewer non possono avere cardinalità 0 (false)

2) Anche per la seconda generalizzazione ho deciso di utilizzare l'accorpamento dei figli nel padre. Questo perché semplificherebbe di gran lunga le operazioni e non ridurrebbe lo spazio occupato e il tempo di accesso. Ritengo che sia la scelta più efficiente in quanto s'eviterebbe di dover introdurre ulteriormente nuove entità e associazioni che complicherebbero il tutto.



Aggiunte Business Rule:

- L'attributo video ha cardinalità (0,1)
- L'attributi live ha cardinalità (0,1)
- Se l'attributo live è true (=1) allora video è per forza false (=0), e viceversa.
- Gli attributi streamer e viewer non possono essere entrambi false (=0)
- Se l'attributo live è true (=1) allora l'attributo orario indica l'inizio della live
- Se l'attributo live è true (=1) allora l'attributo visualizzazioni è NULL
- Se l'attributo video è true (=1) allora l'attributo orario può essere NULL

- Se l'attributo video è true (=1) allora l'attributo media_spettatori è NULL

2.3.3) Eventuale partizionamento/accorpamento di entità o associazioni

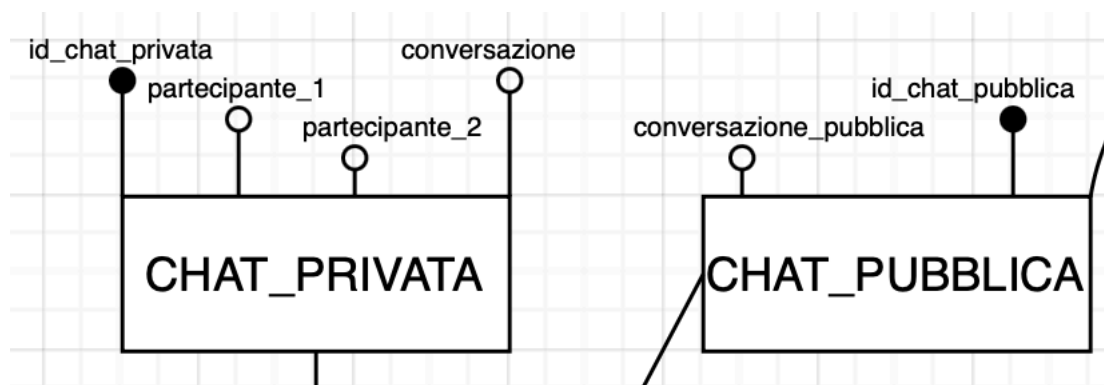
Ho ritenuto che non sia necessario nessun partizionamento o

accorpamento di entità o associazioni.

2.3.4) Eventuale eliminazione degli attributi composti e degli attributi multivalore

- Non sono presenti attributi composti all'interno del diagramma.
- Attributi multivalore:
 1. UTENTE → chat (0,N)
 2. UTENTE → lista_preferiti (0,N)
 3. CANALE → social (0,N)
 4. CONTENUTI → tag (1,N)
 5. UTENTE → calendario(attributo composto)

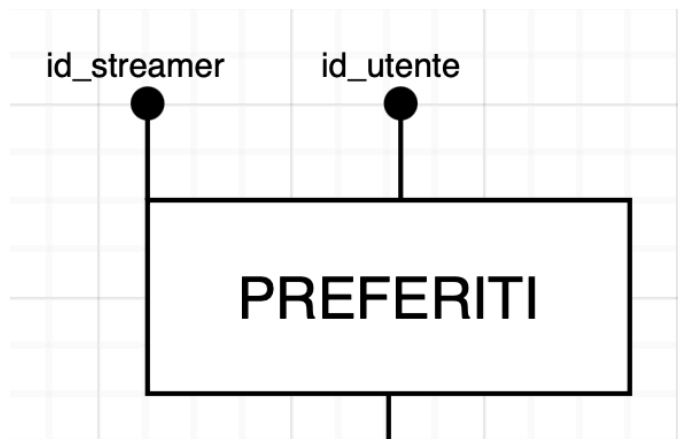
1) Chat viene suddiviso in due per comodità ed efficienza



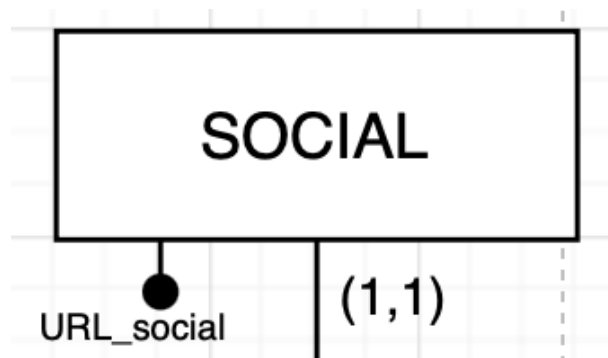
Aggiunta Business Rules:

1. Partecipante_1 e Partecipante_2 fanno riferimento a due utenti, dunque non possono essere entrambi lo stesso utente
2. Conversazione è l'insieme dei messaggi, ogni volta che un utente manda un messaggio in realtà modifica l'attributo 'conversazione' andando ad aggiungere alla fine il suo nuovo messaggio/risposta.

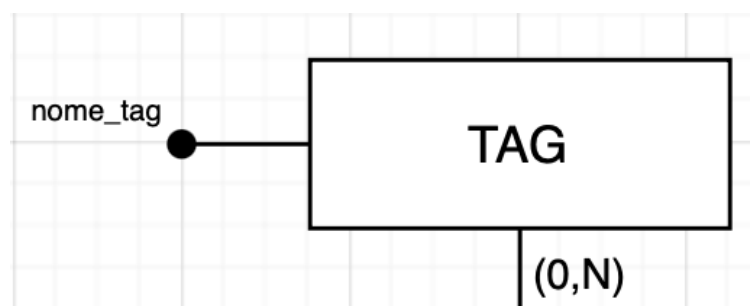
2) Lista preferiti diventa PREFERITI



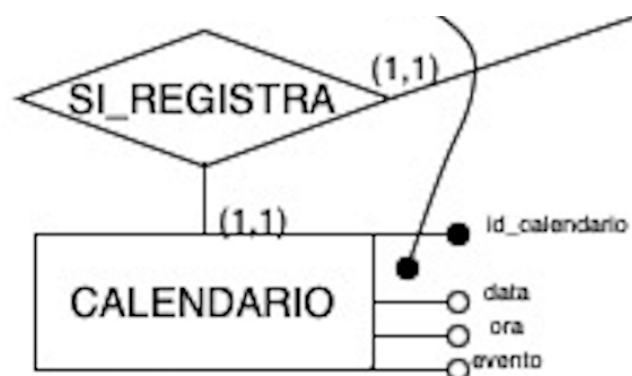
3) Social diventa SOCIAL



4) Tag diventa TAG



5) Calendario diventa CALENDARIO



2.3.5) Eventuale scelta degli identificatori principali

I seguenti identificatori sono stati scelti in modo tale da rappresentare nel modo più semplice la base di dati. La maggior parte di essi sono attributi richiesti dalla traccia, mentre altri sono stati aggiunti o modificati per una maggiore leggibilità e correttezza.

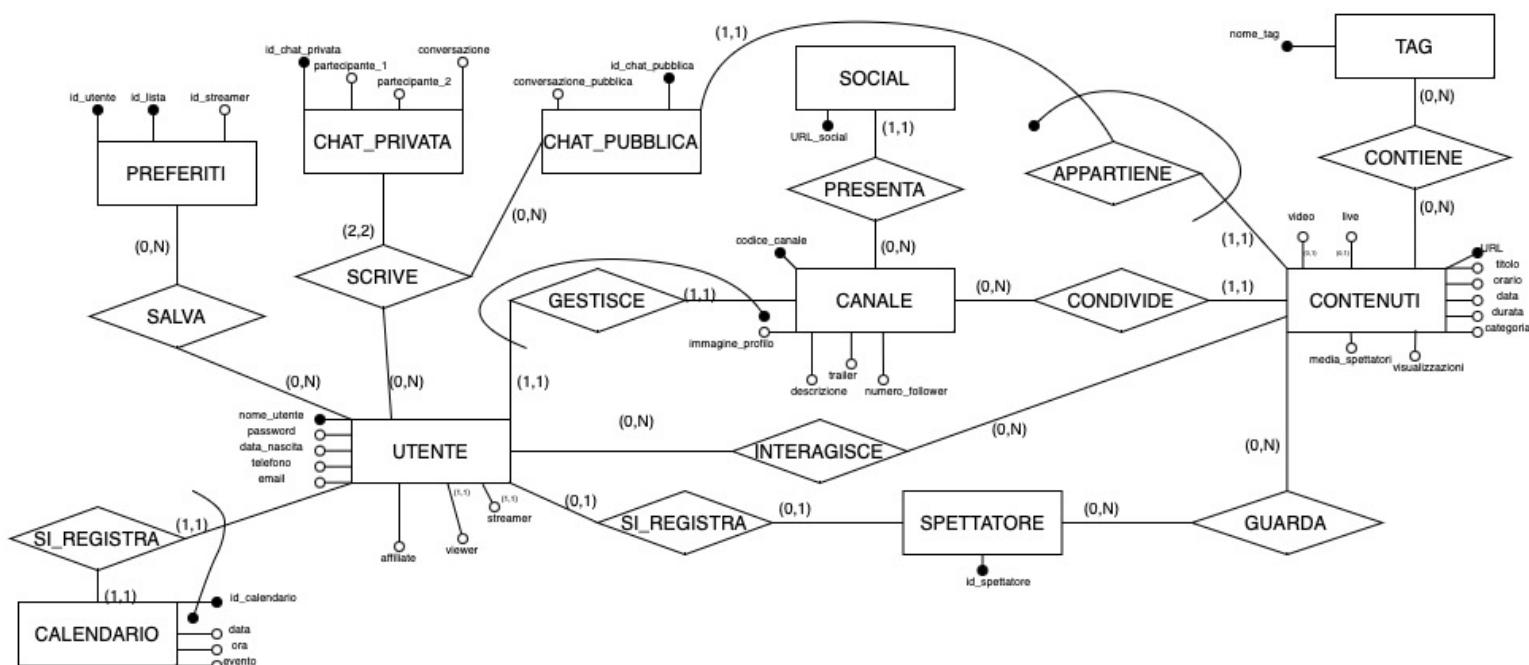
NOME ENTITA'	IDENTIFICATORE PRINCIPALE
UTENTE	nome_utente
PREFERITI	id_lista, id_utente
CHAT_PRIVATA	id_chat_privata
CHAT_PUBBLICA	id_chat_pubblica
CANALE	codice_canale - nome_utente
SOCIAL	URL_social
SPETTATORE	id_spettatore
CONTENUTI	URL
TAG	nome_tag
CALENDARIO	id_calendario - nome_utente

**id_streamer e id_utente sono chiave univoca per PREFERITI

** id_streamer fa riferimento al "nome_utente" di uno streamer attivo

** id_utente fa riferimento al "proprietario" della lista, quindi al nome_utente di UTENTE

2.4) Schema E/R ristrutturato + Business Rule



BUSINESS RULE:

- Per registrarsi gli utenti devono inserire tutti i dati e scegliere se mettere il telefono e/o l'email
- Quando uno streamer raggiunge almeno 500 minuti trasmessi, una media superiore/uguale a 3 spettatori simultanei ed un minimo di 50 follower allora può diventare affiliate
- I viewer ottengono privilegi se effettuano la subscription
- Gli utenti possono seguire lo streamer e creare dirette
- Le live possono diventare o no dei video
- Un video di breve durata è una clip (<30sec)
- Gli utenti non possono donare bit se il loro portafoglio è vuoto o il saldo non è sufficiente
- Gli streamer possono programmare le live aggiungendo gli eventi sul calendario
- Uno spettatore può registrarsi alla piattaforma e diventare un utente, ma un utente non può essere uno spettatore.

Nuove Business Rule aggiunte:

- L' attributo streamer ha cardinalità (1,1)
- L' attributo viewer ha cardinalità (1,1)
- Gli attributi streamer e viewer non possono avere cardinalità 0 (false)
- L' attributo video ha cardinalità (0,1)
- L' attributi live ha cardinalità (0,1)
- Se l'attributo live è true (=1) allora video è per forza false (=0), e viceversa.
- Gli attributi streamer e viewer non possono essere entrambi false (=0)
- Se l'attributo live è true (=1) allora l'attributo orario indica l'inizio della live
- Se l'attributo live è true (=1) allora l'attributo visualizzazioni è NULL
- Se l'attributo video è true (=1) allora l'attributo orario può essere NULL

- Se l'attributo video è true (=1) allora l'attributo media_spettatori è NULL
- Partecipante_1 e Partecipante_2 fanno riferimento a due utenti, dunque non possono essere entrambi lo stesso utente
- Conversazione è l'insieme dei messaggi, ogni volta che un utente manda un messaggio in realtà modifica l'attributo 'conversazione' andando ad aggiungere alla fine il suo nuovo messaggio/risposta.

REGOLE DI DERIVAZIONE:

- Per ogni live viene calcolato il numero medio di spettatori
- Per ogni video viene calcolato il numero di visualizzazioni
- Per gli streamer vengono calcolati il numero di live effettuate, il numero di minuti trasmessi ed il numero medio di spettatori simultanei

2.5) Schema relazionale

- UTENTE(nome_utente, password, data_nascita, telefono, email, media_spettatori, affiliate, calendario, viewer, streamer)
- PREFERITI(id_lista, id_utente, id_streamer)
- CALENDARIO(id_calendario, id_utente, data, ora, evento)
- CHAT_PRIVATA(id_chat_privata, partecipante_1, partecipante_2, conversazione)
- CHAT_PUBBLICA(id_chat_pubblica, contenuti, conversazione_pubblica)
- CANALE(codice_canale, utente, descrizione, trailer, numero_follower)
- SOCIAL(URL_social, canale)
- SPETTATORE(id_spettatore)
- CONTENUTI(URL, canale, titolo, orario, data, durata, categoria, visualizzazioni, media_spettatori, live, video)
- TAG(nome_tag)
- SALVA(preferito, utente)
- SCRIVE(chat_privata, chat_pubblica, utente)

- INTERAGISCE(utente, contenuto)
- SI_REGISTRA(utente, spettatore)
- GUARDA (spettatore, contenuto)
- CONTIENE(tag, contenuto)

- SALVA(preferito) referencia PREFERITI(nome_streamer)
- SALVA(utente) referencia UTENTE(nome_utente)
- SCRIVE(chat_privata) referencia CHAT_PRIVATA(id_chat_privata)
- SCRIVE(chat_pubblica) referencia CHAT_PUBBLICA(id_chat_pubblica)
- SCRIVE(utente) referencia UTENTE(nome_utente)
- CANALE(utente) referencia UTENTE(nome_utente)
- Vincolo UNIQUE su CANALE(utente)
- INTERAGISCE(utente) referencia UTENTE(nome_utente)
- INTERAGISCE(contenuto) referencia CONTENUTI(URL)
- SI_REGISTRA(utente) referencia UTENTE(nome_utente)
- SI_REGISTRA(spettatore) referencia SPETTATORE(id_spettatore)
- Vincolo UNIQUE su SI_REGISTRA(spettatore)
- CALENDARIO(id_utente) referencia UTENTE(nome_utente)
- SOCIAL(canale) referencia CANALE(codice_canale)
- CONTENUTI(canale) referencia CANALE(codice_canale)
- GUARDA(spettatore) referencia SPETTATORE(id_spettatore)
- GUARDA(contenuto) referencia CONTENUTI(URL)
- CONTIENE(tag) referencia TAG(nome_tag)
- CONTIENE(contenuto) referencia CONTENUTI(URL)
- CHAT_PUBBLICA(contenuto) referencia CONTENUTI(URL)
- Vincolo UNIQUE su CHAT_PUBBLICA(contenuto)

Vincoli d'integrità referenziale:

- Un utente può salvare tra i preferiti più streamer (viene concesso di poter salvare anche se stessi per semplicità)

- Nell'entità chat_privata gli attributi partecipante_1 e partecipante_2 fanno riferimento a degli utenti, ma non possono essere uguali
- Gli attributi viewer e streamer non possono essere nulli
- Gli attributi video e live possono essere nulli ma non entrambi

Implementazione

3.1) DDL di creazione del database

```

create table UTENTE(
    nome_utente varchar(20) primary key,
    password_ varchar(30) not null,
    data_nascita date not null,
    telefono numeric(10) CHECK (telefono > 999999999),
    email varchar(30),
    affiliate boolean,
    viewer boolean CHECK(viewer IN (true, false)),
    streamer boolean CHECK(streamer IN (true, false)),
    CONSTRAINT Utente_Registration_Check
        CHECK ((telefono IS NOT NULL OR Email IS NOT NULL) AND nome_utente IS NOT
NULL AND password_ IS NOT NULL
        AND data_nascita IS NOT NULL)
);

create table PREFERITI(
    id_streamer varchar(20) unique references UTENTE(nome_utente) ON DELETE
CASCADE ON UPDATE CASCADE,
    id_utente varchar(20) unique references UTENTE(nome_utente) ON DELETE CASCADE
ON UPDATE CASCADE,
    primary key(id_streamer, id_utente)
);

create table CALENDARIO(
    id_calendario numeric(20),

```

```

        id_utente varchar(20) unique references UTENTE(nome_utente) ON DELETE CASCADE
ON UPDATE CASCADE,
        data_date,
        ora time,
        evento varchar(30),
        primary key(id_calendario, id_utente)
);

```

```

create table CANALE(
        codice_canale numeric(8) primary key,
        utente_canale varchar(20) unique references UTENTE(nome_utente) ON DELETE SET
NULL ON UPDATE CASCADE,
        descrizione varchar(2000),
        trailer boolean CHECK(trailer IN (true, false)),
        numero_follower numeric(10)
);

```

```

create table CONTENUTI(
        URL varchar(50) primary key,
        canale numeric(8) references CANALE(codice_canale) ON DELETE NO ACTION ON
UPDATE CASCADE,
        titolo varchar(30) not null,
        orario time not null,
        data_date not null,
        durata numeric(5) not null,
        categoria varchar(20) not null,
        visualizzazioni numeric(10),
        live boolean CHECK(live IN (true, false)),
        video boolean CHECK(video IN (true, false)),
        CONSTRAINT Live_Video_Check
        CHECK ((live = true AND video = false) OR (live = false AND video = true)),
        CONSTRAINT Live_Attributes_Check
        CHECK ((live = true AND orario IS NOT NULL AND visualizzazioni IS NULL) OR (live =
false AND orario IS NULL
        AND visualizzazioni IS NOT NULL))
);

```

);

```
create table CHAT_PRIVATA(  
    id_chat_privata numeric(10) primary key,  
    partecipante_1 varchar(20) not null references UTENTE(nome_utente) ON DELETE SET  
NULL ON UPDATE CASCADE,  
    partecipante_2 varchar(20) not null references UTENTE(nome_utente) ON DELETE SET  
NULL ON UPDATE CASCADE,  
    conversazione TEXT,  
    CONSTRAINT Distinct_Participants_Check  
CHECK (partecipante_1 <> partecipante_2)  
);
```

```
create table CHAT_PUBBLICA(  
    id_chat_pubblica numeric(10) primary key,  
    contenuti varchar(50) references CONTENUTI(URL) ON DELETE NO ACTION ON  
UPDATE CASCADE,  
    conversazione_pubblica TEXT  
);
```

```
create table SOCIAL(  
    URL_social varchar(50) primary key,  
    canale numeric(8) references CANALE(codice_canale) ON DELETE NO ACTION ON  
UPDATE CASCADE  
);
```

```
create table SPETTATORE(  
    id_spettatore numeric(20) primary key  
);
```

```
create table TAG(  
    nome_tag varchar(20) primary key  
);
```

```
create table SALVA(  

```

```

        preferito varchar(20) references PREFERITI(id_streamer) ON DELETE CASCADE ON
UPDATE CASCADE,
        utente varchar(20) references UTENTE(nome_utente) ON DELETE CASCADE ON
UPDATE CASCADE,
        primary key(preferito, utente)
);

```

create table SCRIVE(

```

        chat_privata numeric(10) references CHAT_PRIVATA(id_chat_privata)
ON DELETE SET NULL ON UPDATE CASCADE,
        chat_pubblica numeric(10) references CHAT_PUBBLICA(id_chat_pubblica)
ON DELETE SET NULL ON UPDATE CASCADE,
        utente varchar(20) references UTENTE(nome_utente),
        primary key(chat_privata, chat_pubblica, utente)
);

```

create table INTERAGISCE(

```

        utente varchar(20) references UTENTE(nome_utente) ON DELETE SET NULL ON
UPDATE CASCADE,
        contenuto varchar(50) references CONTENUTI(URL) ON DELETE NO ACTION ON
UPDATE CASCADE,
        primary key(utente, contenuto)
);

```

create table SI_REGISTRA(

```

        utente varchar(20) references UTENTE(nome_utente) ON DELETE CASCADE ON
UPDATE CASCADE,
        spettatore numeric(20) primary key unique references SPETTATORE(id_spettatore)
ON DELETE SET NULL ON UPDATE CASCADE
);

```

create table GUARDA(

```

        spettatore numeric(20) references SPETTATORE(id_spettatore) ON DELETE SET NULL
ON UPDATE CASCADE,
        contenuto varchar(50) references CONTENUTI(URL) ON DELETE NO ACTION ON
UPDATE CASCADE,

```

```

        primary key(spettatore, contenuto)
    );

    create table CONTIENE(
        tag varchar(20) references TAG(nome_tag) ON DELETE CASCADE ON UPDATE
        CASCADE,
        contenuto varchar(50) references CONTENUTI(URL) ON DELETE NO ACTION ON
        UPDATE CASCADE,
        primary key(tag, contenuto)
    );

```

3.2) DML di popolamento di tutte le tabelle del database

```

INSERT INTO UTENTE (nome_utente, password_, data_nascita, telefono, email, affiliate,
viewer, streamer)
VALUES ('user1', 'password1', '2000-01-01', 1234567890, 'user1@example.com', false,
true, false);

```

```

INSERT INTO UTENTE (nome_utente, password_, data_nascita, telefono, email, affiliate,
viewer, streamer)
VALUES ('user2', 'password2', '2001-01-01', 2345678901, 'user2@example.com', false,
false, true);

```

```

INSERT INTO UTENTE (nome_utente, password_, data_nascita, telefono, email, affiliate,
viewer, streamer)
VALUES ('user3', 'password3', '2002-01-01', 3456789012, 'user3@example.com', false,
false, false);

```

```

INSERT INTO PREFERITI (id_streamer, id_utente, id_lista)
VALUES ('user1', 'user2', 1);

```

```

INSERT INTO PREFERITI (id_streamer, id_utente, id_lista)
VALUES ('user2', 'user3', 2);

```

```

INSERT INTO PREFERITI (id_streamer, id_utente, id_lista)

```



```
VALUES ('user3', 'user1', 3);
```

```
INSERT INTO CALENDARIO (id_calendario, id_utente, data_, ora, evento)  
VALUES (1, 'user1', '2023-12-08', '10:00:00', 'Event 1');
```

```
INSERT INTO CALENDARIO (id_calendario, id_utente, data_, ora, evento)  
VALUES (2, 'user2', '2023-12-09', '11:00:00', 'Event 2');
```

```
INSERT INTO CALENDARIO (id_calendario, id_utente, data_, ora, evento)  
VALUES (3, 'user3', '2023-12-10', '12:00:00', 'Event 3');
```

```
INSERT INTO CANALE (codice_canale, utente_canale, descrizione, trailer,  
numero_follower)  
VALUES (1, 'user1', 'Description 1', true, 100);
```

```
INSERT INTO CANALE (codice_canale, utente_canale, descrizione, trailer,  
numero_follower)  
VALUES (2, 'user2', 'Description 2', false, 200);
```

```
INSERT INTO CANALE (codice_canale, utente_canale, descrizione, trailer,  
numero_follower)  
VALUES (3, 'user3', 'Description 3', true, 300);
```

```
INSERT INTO CONTENUTI (URL, canale, titolo, orario, data_, durata, categoria,  
visualizzazioni, live, video)  
VALUES ('URL1', 1, 'Title 1', '10:00:00', '2023-12-08', 5, 'Category 1', NULL, true, false);
```

```
INSERT INTO CONTENUTI (URL, canale, titolo, orario, data_, durata, categoria,  
visualizzazioni, live, video)  
VALUES ('URL2', 2, 'Title 2', '11:00:00', '2023-12-09', 10, 'Category 2', 2000, false, true);
```

```
INSERT INTO CONTENUTI (URL, canale, titolo, orario, data_, durata, categoria,  
visualizzazioni, live, video)  
VALUES ('URL3', 3, 'Title 3', '12:00:00', '2023-12-10', 15, 'Category 3', NULL, true, false);
```

```
INSERT INTO CHAT_PRIVATA (id_chat_privata, partecipante_1, partecipante_2,  
conversazione)
```

```
VALUES (1, 'user1', 'user2', 'Hello, User2!');
```

```
INSERT INTO CHAT_PRIVATA (id_chat_privata, partecipante_1, partecipante_2,  
conversazione)
```

```
VALUES (2, 'user2', 'user3', 'Hello, User3!');
```

```
INSERT INTO CHAT_PRIVATA (id_chat_privata, partecipante_1, partecipante_2,  
conversazione)
```

```
VALUES (3, 'user3', 'user1', 'Hello, User1!');
```

```
INSERT INTO CHAT_PUBBLICA (id_chat_pubblica, contenuti)
```

```
VALUES (1, 'URL1');
```

```
INSERT INTO CHAT_PUBBLICA (id_chat_pubblica, contenuti)
```

```
VALUES (2, 'URL2');
```

```
INSERT INTO CHAT_PUBBLICA (id_chat_pubblica, contenuti)
```

```
VALUES (3, 'URL3');
```

```
INSERT INTO SOCIAL (URL_social, canale)
```

```
VALUES ('URL_social1', 1);
```

```
INSERT INTO SOCIAL (URL_social, canale)
```

```
VALUES ('URL_social2', 2);
```

```
INSERT INTO SOCIAL (URL_social, canale)
```

```
VALUES ('URL_social3', 3);
```

```
INSERT INTO SPETTATORE (id_spettatore)
```

```
VALUES (1);
```

```
INSERT INTO SPETTATORE (id_spettatore)
```

```
VALUES (2);
```

```
INSERT INTO SPETTATORE (id_spettatore)
VALUES (3);
```

```
INSERT INTO TAG (nome_tag)
VALUES ('Tag1');
```

```
INSERT INTO TAG (nome_tag)
VALUES ('Tag2');
```

```
INSERT INTO TAG (nome_tag)
VALUES ('Tag3');
```

```
INSERT INTO SALVA (preferito, utente)
VALUES ('user1', 'user2');
```

```
INSERT INTO SALVA (preferito, utente)
VALUES ('user2', 'user3');
```

```
INSERT INTO SALVA (preferito, utente)
VALUES ('user3', 'user1');
```

```
INSERT INTO SCRIVE (chat_privata, chat_pubblica, utente)
VALUES (1, 1, 'user1');
```

```
INSERT INTO SCRIVE (chat_privata, chat_pubblica, utente)
VALUES (2, 2, 'user2');
```

```
INSERT INTO SCRIVE (chat_privata, chat_pubblica, utente)
VALUES (3, 3, 'user3');
```

```
INSERT INTO INTERAGISCE (utente, contenuto)
VALUES ('user1', 'URL1');
```

```
INSERT INTO INTERAGISCE (utente, contenuto)
VALUES ('user2', 'URL2');
```

```
INSERT INTO INTERAGISCE (utente, contenuto)
VALUES ('user3', 'URL3');
```

```
INSERT INTO SI_REGISTRA (utente, spettatore)
VALUES ('user1', 1);
```

```
INSERT INTO SI_REGISTRA (utente, spettatore)
VALUES ('user2', 2);
```

```
INSERT INTO SI_REGISTRA (utente, spettatore)
VALUES ('user3', 3);
```

```
INSERT INTO GUARDA (spettatore, contenuto)
VALUES (1, 'URL1');
```

```
INSERT INTO GUARDA (spettatore, contenuto)
VALUES (2, 'URL2');
```

```
INSERT INTO GUARDA (spettatore, contenuto)
VALUES (3, 'URL3');
```

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

```
UPDATE CANALE
SET descrizione = 'Updated Description 1'
WHERE utente_canale = 'user1';
```

```
UPDATE UTENTE
SET email = 'newuser2@example.com'
WHERE nome_utente = 'user2';
```

```
UPDATE CONTENUTI  
SET titolo = 'Updated Title 2'  
WHERE URL = 'URL2';
```

```
DELETE FROM PREFERITI  
WHERE id_streamer = 'user1' AND id_utente = 'user2';
```

```
DELETE FROM CALENDARIO  
WHERE id_utente = 'user2';
```

```
DELETE FROM SOCIAL  
WHERE URL_social = 'URL3';
```