



**UNIVERSITÀ
DI TORINO**

Laboratorio Basi di Dati 2022/2023

Relazione del progetto di:

Boggiatto Alex Matr. 946688

Bordis Paolo Alberto Matr. 949351

Caricasulo Francesco Matr. 944173

Richiesta:

Ci veniva richiesto di progettare ed elaborare una base di dati in grado di gestire un sito di streaming video, che permettesse a utenti registrati e non, di poter guardare ciò che veniva trasmesso, di poter interagire con chi era in diretta e anche con altri utenti esterni. Di seguito sono riportati i passaggi che abbiamo effettuato durante il nostro studio.

1. Progettazione Concettuale:

1.1 Requisiti iniziali

Legenda:

Giallo: Entità

Verde: Attributi

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**. 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** 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:

Termini	Descrizione	Sinonimi	Collegamenti
Live	Live streaming fatta dallo streamer e visionata dagli spettatori	Diretta	Streamer, Spettatore
Utente	Utilizzatore della piattaforma di streaming in modalità anonima		Streamer, Spettatore
Spettatore	Utente iscritto alla piattaforma che visualizza le live	viewer	Canale, Live, Utente
Streamer	Utente che produce live e contenuti sulla piattaforma (iscrizione obbligatoria) Possiede un canale	Creatore di Contenuti	Live, Canale, Utente
Canale	Pagina contenente le informazioni e i contenuti dello streamer collegato Schedulati nel calendario	Pagina del canale	Streamer, Spettatore, Calendario, Video
Video	Video caricati sul canale dello streamer, inerenti alle live prodotte		Canale, Live passata, Clip
clip	video di breve durata sulle live trasmesse sul canale		Video
Calendario	documento presente sul canale contenente le informazioni delle live future		Canale
Portafoglio	Saldo di bit posseduti dall'utente		Utente
Categoria	Tipo di argomenti trattati all'interno della live		
Tag	Strumento utilizzato per la ricerca della live		
Messaggio	Testo di parole inviato da un utente ad un altro utente		Utente

1.3 Requisiti Rivisti e Ristrutturati:

Dominio: TWITCH

Si vuole realizzare un servizio che permette di fare live streaming su vari argomenti e di poterne seguire, deve contenere informazioni riguardo gli spettatori, gli streamer, calcolare le informazioni riguardanti la diretta eseguita dallo streamer e le informazioni del suo canale.

Utente:

- Spettatore anonimo che una volta eseguita la registrazione può suddividersi in "Spettatore" o "Streamer".

Streamer:

- Gli streamer sono degli utenti registrati, che decidono di avviare delle live, uno streamer diventa "affiliate" se rispetta i seguenti requisiti:
 - Il Numero di minuti trasmesso è maggiore o uguale di 500.
 - Il Numero medio di spettatori è maggiore o uguale a 3.
 - Il Numero di follower è maggiore o uguale a 50.

Spettatore:

- Gli spettatori sono persone che hanno eseguito la registrazione al servizio e devono fornire:
 - username
 - password
 - email
 - telefono
 - data di nascita
- Lo Spettatore può aggiungere alcuni canali alla sua lista di seguiti
- Può decidere di abbonarsi ad alcuni canali

Messaggio:

- Messaggio che un utente scrive verso un'altro utente.

Canale:

- Un canale è posseduto da uno Streamer ed è possibile specificare:
 - Social associati
 - immagine profilo
 - trailer
- Su un canale vengono caricati i video e le clip
- Sul canale viene memorizzato il numero medio di follower

Live:

- La live viene eseguita sul canale dallo Streamer e ha i seguenti dati:
 - titolo
 - durata
- Per ogni live vengono memorizzati il numero medio di spettatori

Video:

- Il video è una live conclusa e caricata sul canale, ha i seguenti dati
 - categoria
 - argomento
 - tag
 - durata
 - titolo
 - numero visualizzazioni

Clip:

- La clip è un video di breve durata che viene generata dai video e dalle live, ha i seguenti dati:
 - titolo
 - durata
 - numero visualizzazioni

Categoria:

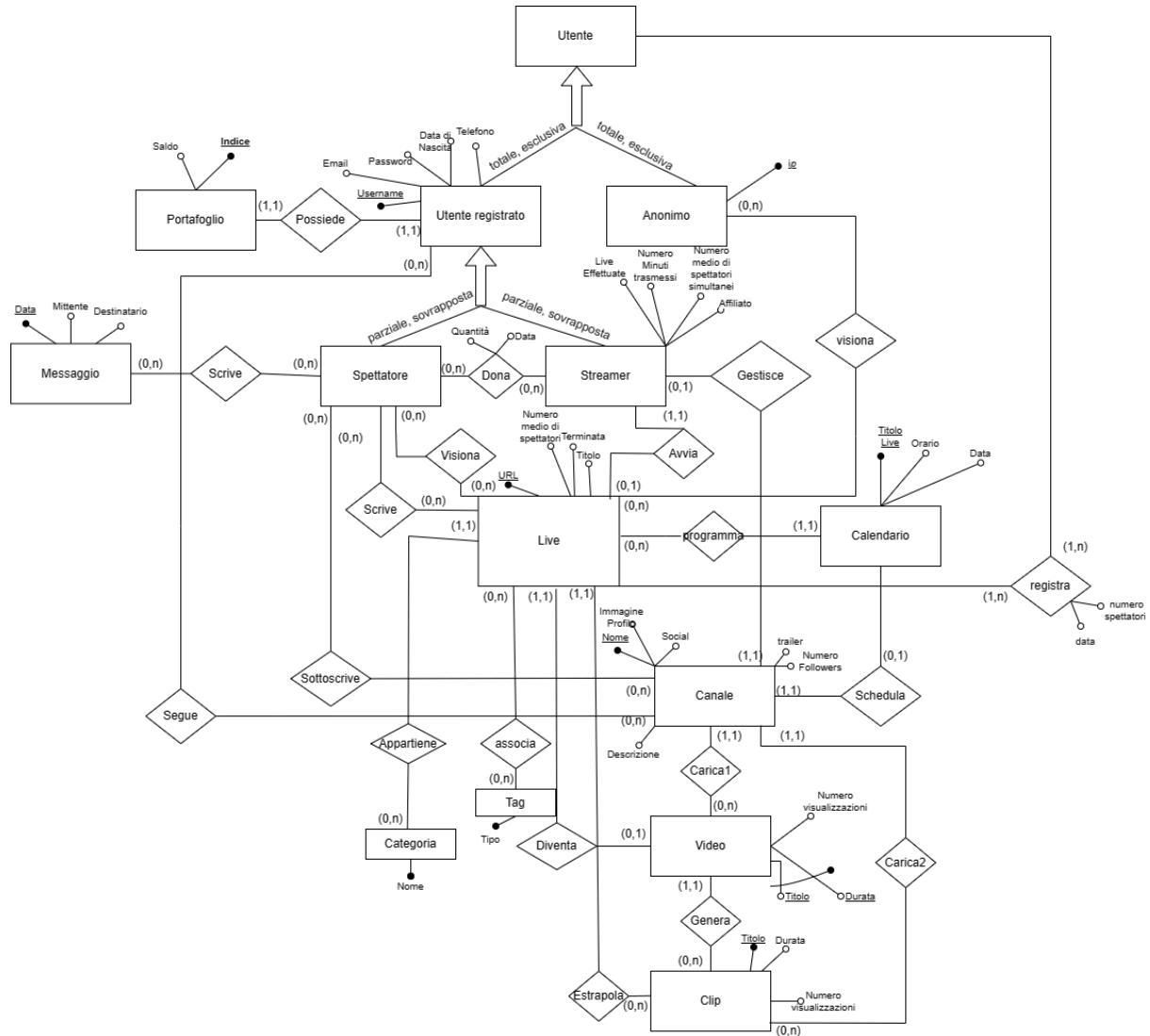
- Una categoria appartiene ad una live e può variare nel tempo, possiede un nome

Tag:

- Uno o più tag appartengono ad una live e possiedono un tipo

1.4 Schema ER + Business Rules:

La progettazione concettuale è il primo step nel quale il database comincia a prendere forma.



Business Rules:

Condizioni:

1. Streamer è affiliato se il numero di minuti trasmessi è maggiore o uguale di 500, il numero di spettatori medio è maggiore o uguale a 3, il numero dei followers è maggiore o uguale a 50;
2. Una volta al giorno viene controllato se uno streamer rispetta i requisiti per diventare affiliato;
3. Una volta a settimana viene calcolata la classifica degli streamer più seguiti.
4. Clip è un video di breve durata derivante da un video sul canale o da una live, minore o uguale di 180 secondi;
5. Una live può diventare video se è terminata (attributo "terminata" su live è True);
6. Il numero medio di spettatori in live verrà calcolato facendo la media dei conteggi registrati in un certo periodo all'interno della relazione ricorsiva Registra
7. Il numero medio di spettatori simultanei in streamer viene calcolato sommando il numero medio spettatori delle live trasmesse dallo streamer durante la settimana diviso il numero di minuti trasmessi

2. Progettazione Logica

Lo scopo della progettazione logica è quello di raggiungere un modello relazionale. Preliminarmente alla progettazione è necessario avere però a disposizione la tabella dei volumi e delle frequenze necessarie.

2.1 Tavola dei Volumi

Tabella volumi			
Concetto	Tipo	Volumi	Descrizione
Utente	E	1.000.000	Qualsiasi persona che accede al sito
Utente Registrato	E	600.000	Utente che si registra alla piattaforma
Anonimo	E	400.000	Utente che decide di non iscriversi
Spettatore	E	400.000	Utente che visiona delle live
Streamer	E	200.000	Utente che avvia una live
Canale	E	600.000	Dettagli del canale associato a ciascuno streamer
Live	E	150.000	Contenuto offerto da uno Streamer sul proprio canale
Video	E	200.000	Live conclusa caricata sul canale
Clip	E	250.000	Segmento di live o video dalla breve durata
Categoria	E	250	Tipologia di contenuto offerto dalla live
Tag	E	300.000	Informazioni aggiuntive per ricercare una live
Calendario	E	200.000	Sistema per schedulare le dirette durante la settimana
Portafoglio	E	600.000	Deposito di bit posseduto dagli utenti

Segue	R	1.000.000	Utente che segue un canale
Gestisce	R	200.000	Lo Streamer gestisce un Canale
Avvia	R	150.000	Lo streamer avvia una diretta
Genera	R	200.000	Dal video vengono generate delle clip
Estrapola	R	250.000	Una clip estrapola un segmento della live
Visiona	R	450.000	Uno spettatore che visiona delle live
Visiona	R	400.000	Un utente anonimo che visiona una live
Sottoscrive	R	250.000	Spettatore che si abbona ad un canale
Possiede	R	600.000	Utente che possiede un portafoglio
Scrive	R	400.000	Spettatore che scrive sulla chat della live
Scrive	R	400.000	Spettatore che scrive ad un altro Spettatore
Diventa	R	150.000	Una live che diventa video
Appartiene	R	250	Categoria che appartiene ad una live
Associa	R	300.000	Tag associati alla live
Dona	R	500.000	Utente che dona allo streamer

2.2 Tavola delle Operazioni

Tabella delle operazioni			
Operazioni	Tipo	Frequenza	Descrizione
Registrazione Utente	I	100000/giorno	
Effettuare donazione	I	10000/minuto	
Effettuare un follow per un canale	I	1000/giorno	
Visionare live	I	100000/minuto	
Effettuare schedulazione del calendario del canale	I	1/Settimana	
Interazione chat	I	100/minuto	Utenti che scrivono sulla chat durante una live
Calcolo affiliato	B	1/Giorno	
Calcolo classifica Streamer più seguiti	B	1/Settimana	
Aggiornamenti o Spettatori medi della live	B	100/Giorno	

2.3 Ristrutturazione schema E-R

2.3.1 Analisi delle ridondanze:

Una ridondanza in uno schema E-R è una informazione significativa ma derivabile da altre

In questa fase si decide se eliminare le ridondanze eventualmente presenti o mantenerle

Se si mantiene una ridondanza

- si semplificano alcune interrogazioni, ma
- si appesantiscono gli aggiornamenti
- si occupa maggior spazio

Le possibili ridondanza riguardano

- Attributi derivabili da altri attributi
- Relazioni derivabili dalla composizione di altre relazioni (presenza di cicli)

1. Analisi della ridondanza dell'attributo “numero di spettatori medio” in Live, derivabile dal calcolo della media tra i valori nella relazione “Registra”.

Operazione 1, memorizza il numero di spettatori medio di una live. (1/giorno)

Analisi Dello schema con ridondanza

Concetto	Costrutto	Accessi	Tipo
Live	E	1	S
Registra	R	1	L

Per un totale di 1000 accessi in lettura al giorno e 100 in scrittura, abbiamo 1100 accessi al giorno.

Costo Spazio: Assumendo di utilizzare 4 Byte per memorizzare il numero medio di spettatori ($\sim 2^{12}$ utenti): $4 * 128$ Byte

Analisi Dello schema senza ridondanza

Concetto	Costrutto	Accessi	Tipo
Registra	R	1	L

Per un totale di 1000 accessi in lettura al giorno, abbiamo 1000 accessi al giorno.

Costo Spazio: 0 Byte

2. Analisi della ridondanza dell'attributo "Numero Follower" in Canale, derivabile dal conteggio dell'associazione "Segue".

Analisi Dello schema con ridondanza

Concetto	Costrutto	Accessi	Tipo
Utente registrato	E	1	L
Segue	R	1000	L
Canale	E	100	S

Per un totale di 1000 accessi in lettura 100 in scrittura, abbiamo 1100 accessi al giorno.

Costo Spazio: Assumiamo di usare 4 Byte per memorizzare il numero di follower ($\sim 2^{12}$): 4*128 Byte.

Analisi Dello schema senza ridondanza

Concetto	Costrutto	Accessi	Tipo
Utente registrato	E	1	L
Segue	R	1000	L

Per un totale di 1000 accessi in lettura.

Costo Spazio: 0 Byte

Nella nostra analisi successiva abbiamo deciso di tenere e continuare ad analizzare lo schema senza ridondanze.

2.3.2 Eliminazione delle generalizzazioni:

Obiettivo :

Sostituire le generalizzazioni nello schema ER con strutture alternative, in quanto non direttamente rappresentabili nel modello relazionale.

Tre possibilità:

1. accorpamento delle figlie della generalizzazione nel genitore
2. accorpamento del genitore della generalizzazione nelle figlie
3. sostituzione della generalizzazione con relazioni

- Per la generalizzazione Utente Registrato entità genitore e Spettatore e Streamer entità figlie abbiamo optato per la soluzione numero 1.

- Passi:

- Eliminazione delle entità figlie.
- Accorpamento nel padre di tutti gli attributi specifici delle entità figlie come attributi opzionali.
- Accorpamento nel padre delle relazioni che coinvolgono le entità figlie assegnando cardinalità minima uguale a zero.

L'utente registrato in questo caso si suddivide in Spettatore e Streamer. Quindi è l'utente registrato che può gestire un canale o avviare delle live da streamer, può visionare 0 o più live da spettatore o streamer, può donare a 0 o più streamer da spettatore e sottoscrivere abbonamenti a 0 o più canali da spettatore. Può scrivere a 0 o più utenti registrati o canali da spettatore.

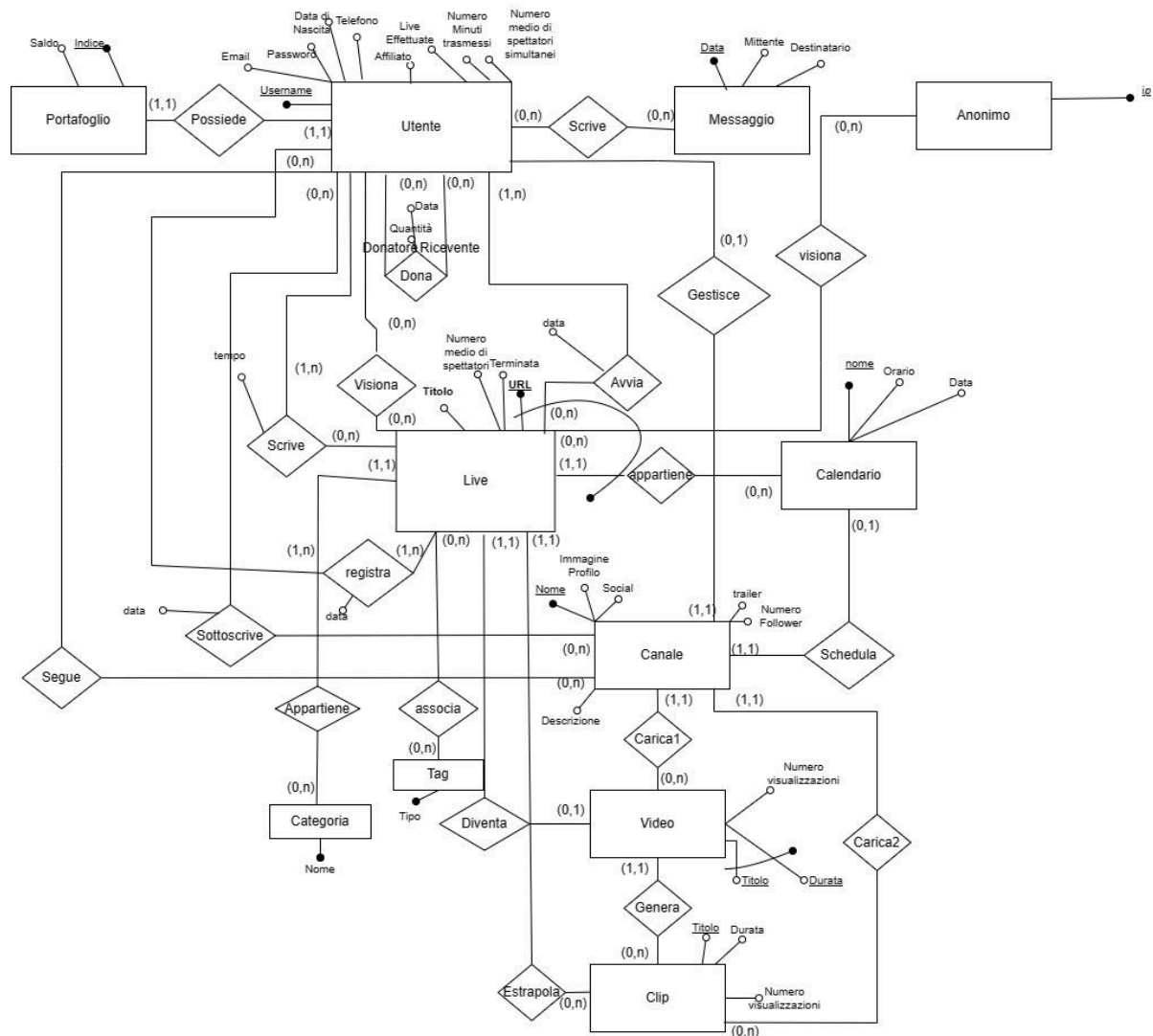
- Per la generalizzazione Utente entità genitore e utente Registrato e Anonimo entità figlie abbiamo optato per la soluzione numero 2, in modo da ridurre gli accessi.

Sono rimaste le entità Utente registrato e Anonimo.

Utente registrato riguarda il caso in cui un utente si è iscritto alla piattaforma.

Anonimo riguarda un utente che non si è iscritto alla piattaforma, e che può visionare le live.

2.4 Schema E-R + business rules



Utente: Accorpa i precedenti Utente Registrato, Streamer e Spettatore

Anonimo: Separato da utente e usato come entità a sé stante.

Business Rules:

Condizioni:

1. Utente è affiliato se il numero di minuti trasmessi è maggiore o uguale di 500, il numero di spettatori medio è maggiore o uguale a 3, il numero dei followers è maggiore o uguale a 50;
2. Una volta al giorno viene controllato se un Utente rispetta i requisiti per diventare affiliato;
3. Una volta a settimana viene calcolata la classifica degli utenti più seguiti.
4. Clip è un video di breve durata derivante da un video sul canale o da una live, minore o uguale di 180 secondi;
5. Una live può diventare video se è terminata (attributo "terminata" su live è True);
6. Il numero medio di spettatori in live verrà calcolato facendo la media dei conteggi registrati in un certo periodo all'interno della relazione ricorsiva Registra
7. Il numero medio di spettatori simultanei in utente viene calcolato sommando il numero medio spettatori delle live trasmesse dallo utente durante la settimana diviso il numero di minuti trasmessi

Vincoli e Regole Aziendali:

Utente:

L'entità utente contiene "Username" come chiave ed altri attributi:

Telefono, email, data di nascita, password che devono essere selezionati obbligatoriamente.

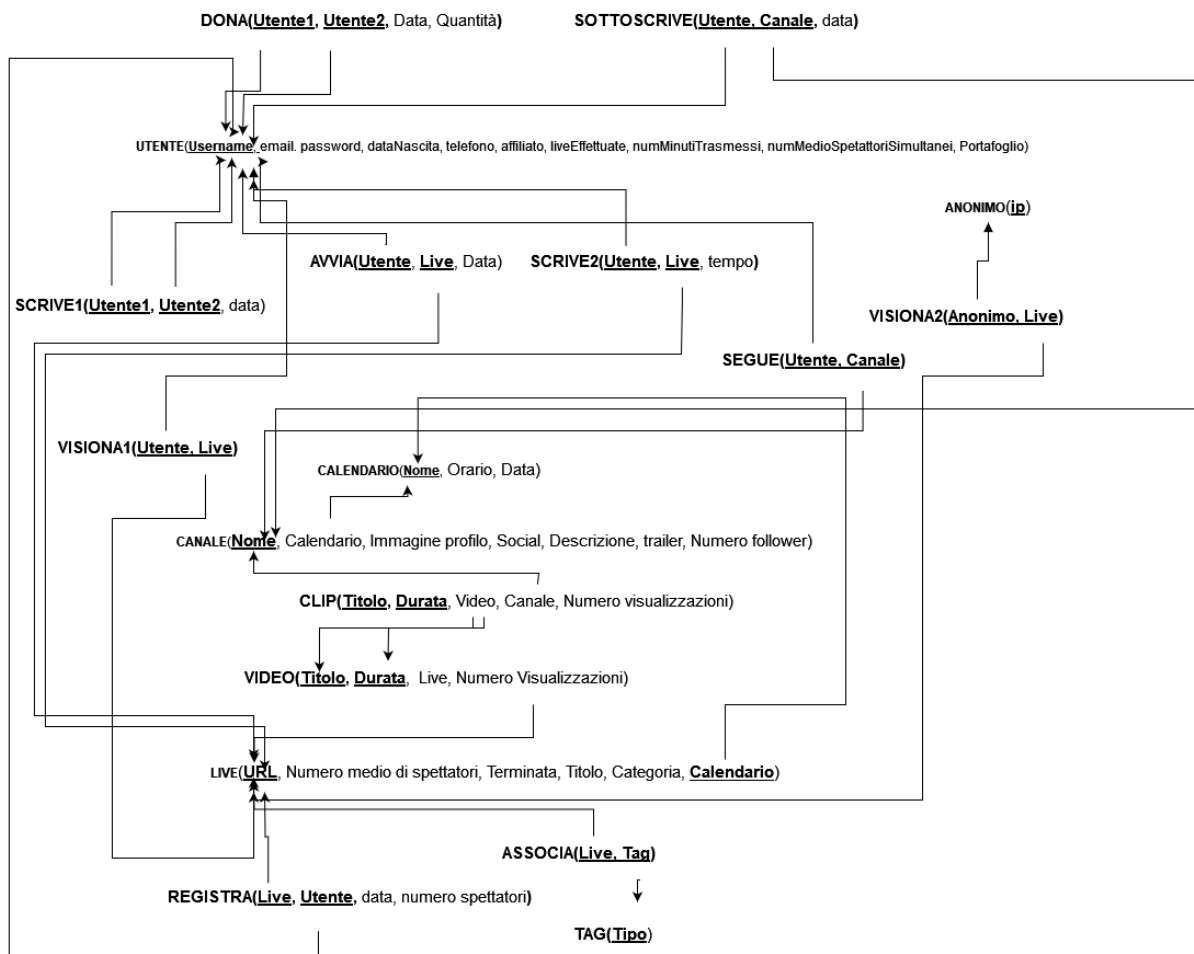
Canale:

La chiave "Nome" di Canale, equivale al Username del Utente proprietario.

Anonimo:

La chiave "ip" in Anonimo, viene autogenerata dal sistema una volta che si accede al sito, ed eliminata quando lo si lascia.

2.5. Schema relazionale



UTENTE(Username, email, password, data di nascita, telefono, affiliato, live effettuate, numero minuti trasmessi, numero medio di spettatori simultanei, Portafoglio)

Utente(Portafoglio) referencia Portafoglio(nome)

ANONIMO(ip)

LIVE(URL, Calendario, Numero medio di spettatori, Terminata, Titolo, Categoria)

Live(Calendario) referencia Calendario(Nome)

CALENDARIO(Nome, Orario, Data)

PORTAFOGLIO(Indice, Saldo)

CATEGORIA(Nome)

CANALE(Nome, Calendario, Immagine profilo, Social, Descrizione, trailer, Numero follower)

Canale(Calendario) referencia Calendario(nome)

Vincolo Unique Canale(Calendario)

TAG(Tipo)

VIDEO(Titolo, Durata, Numero Visualizzazioni, Live, Canale)

Video(Live) referencia Live(URL)

Video(Canale) Referenzia Canale(Nome)

Vincolo unique Video(Live)

CLIP(Titolo, Durata, Video, Canale, Numero visualizzazioni)

Clip(Video) referencia Video(Titolo, durata)

Clip(Canale) referencia Canale(Nome)

SCRIVE1(Utente1, Utente2, data)

Scrive1(Utente1) referencia Utente(Username)

Scrive1(Utente2) referencia Utente(Username)

SCRIVE2(Utente, Live, tempo)

Scrive1(Utente) referencia Utente(Username)

Scrive1(Live) referencia Live(URL)

DONA(Utente1, Utente2, Data, Quantità)

Dona(Utente1) referencia Utente(Username)

Dona(Utente2) referencia Utente(Username)

VISIONA1(Utente, Live)

Visiona1(Utente) referencia Utente(Username)

Visiona1(Live) referencia Live(URL)

VISIONA2(Anonimo, Live)

Visiona2(Anonimo) referencia Anonimo(ip)

Visiona2(Live) referencia Live(URL)

SOTTOSCRIVE(Utente, Canale, data)

Sottoscrive(Utente) referencia Utente(Username)

Sottoscrive(Canale) referencia Canale(Nome)

SEGUE(Utente, Canale)

Segue(Utente) referencia Utente(Username)

Sottoscrive(Canale) referencia Canale(Nome)

REGISTRA(Live, Utente, data, numero spettatori)

Registra(Live) referencia Live(URL)

Registra(Utente) referencia Utente(Username)

ASSOCIA(Live, Tag)

Associa(Live) referencia Live(URL)

Associa(Tag) referencia Tag(Tipo)

AVVIA(Live, Utente, data)

Avvia(Live) referencia Live(URL)

Avvia(Utente) referencia Utente(Username)

3. Implementazione

3.1. DDL di creazione del database

```
CREATE TABLE CATEGORIA(  
    Nome VARCHAR(10) NOT NULL,  
  
    PRIMARY KEY (Nome)  
)
```

```
CREATE TABLE PORTAFOGLIO(  
    Indice int(11) NOT NULL,  
    Saldo Float DEFAULT 0,  
  
    PRIMARY KEY (Indice)  
)
```

```
CREATE TABLE CALENDARIO(  
    Nome VARCHAR(30) NOT NULL,  
    Orario TIME NOT NULL,  
    Datac Date NOT NULL,  
  
    PRIMARY KEY (Nome)  
)
```

```
CREATE TABLE UTENTE(  
    Username varchar(30) NOT NULL,  
    Email varchar(100) NOT NULL,  
    Pass varchar(100) NOT NULL,  
    DataN date NOT NULL,  
    Telefono decimal(10,0) DEFAULT NULL,  
    Affiliato tinyint(1) DEFAULT 0,  
    NumeroMin int(11) DEFAULT 0,  
    NumeroMedSpett int(11) DEFAULT 0,  
    Portafoglio int(11) NOT NULL  
  
    PRIMARY KEY (Username)  
)
```

```
CREATE TABLE LIVE(  
    Indirizzo VARCHAR(50) NOT NULL,  
    Calendario VARCHAR(30),  
    Titolo VARCHAR(30) NOT NULL,  
    NumSpettatori INT DEFAULT 0,  
    Terminata BOOLEAN DEFAULT False,  
    Categoria VARCHAR(10) NOT NULL,  
  
    PRIMARY KEY (Indirizzo),  
    FOREIGN KEY (Calendario) references Calendario(Nome),
```

```
FOREIGN KEY (Categoria) references Categoria(Nome),  
)
```

```
CREATE TABLE TAG(  
    Nome VARCHAR(20) NOT NULL,  
    PRIMARY KEY (Nome)  
)
```

```
CREATE TABLE VIDEO(  
    Titolo VARCHAR(30) NOT NULL,  
    Durata TIME NOT NULL,  
    NumeroVisite INT DEFAULT 0,  
    Live VARCHAR(50) NOT NULL,  
    Canale VARCHAR(50) NOT NULL,  
  
    PRIMARY KEY (Titolo, Durata),  
    FOREIGN KEY (Canale) references Canale(Nome)  
    FOREIGN KEY (Live) references Live(Indirizzo)  
)
```

```
CREATE TABLE CANALE(  
    Nome VARCHAR(30) NOT NULL,  
    Calendario VARCHAR(30) NOT NULL,  
    Immagine URL NULL,  
    Social URL NULL,  
    Descrizione VARCHAR(100) NULL,  
    Trailer URL NULL,  
    numFollow INT DEFAULT 0  
  
    PRIMARY KEY (Nome)  
)
```

```
CREATE TABLE CLIP(  
    Titolo VARCHAR(30) NOT NULL,  
    Durata TIME NOT NULL,  
    NumeroVisite INT DEFAULT 0,  
    Video VARCHAR(30) NOT NULL,  
    Canale VARCHAR(30) NOT NULL,  
  
    PRIMARY KEY (Titolo, Durata),  
    FOREIGN KEY (Video) references Video(Titolo),
```

```

    FOREIGN KEY (Canale) references Canale(Nome)
)

CREATE TABLE ANONIMO(
    Ip INT NOT NULL,

    PRIMARY KEY (Ip)
)

CREATE TABLE SCRIVE1(
    Utente1 VARCHAR(30) NOT NULL,
    Utente2 VARCHAR(30) NOT NULL,
    Data1 DATE,

    FOREIGN KEY(Utente1) references Utente(Username),
    FOREIGN KEY(Utente2) references Utente(Username),
)

CREATE TABLE SCRIVE2(
    Utente VARCHAR(30) NOT NULL,
    Live VARCHAR(30) NOT NULL,
    Tempo TIME,

    FOREIGN KEY (Utente) references Utente(Username),
    FOREIGN KEY (Live) references Live(URL)
)

CREATE TABLE DONA(
    Utente1 VARCHAR(30) NOT NULL,
    Utente2 VARCHAR(30) NOT NULL,
    Data1 DATE,
    valore FLOAT,

    FOREIGN KEY (Utente1) references Utente(Username),
    FOREIGN KEY (Utente2) references Utente(Username),
)

CREATE TABLE VISIONA1(
    Utente VARCHAR(30) NOT NULL,
    Live VARCHAR(30) NOT NULL,

```

```

    FOREIGN KEY (Utente) references Utente(Username),
    FOREIGN KEY (Live) references Live(URL)
)

CREATE TABLE VISIONA2(
    Anonimo VARCHAR(30) NOT NULL,
    Live VARCHAR(30) NOT NULL,

    FOREIGN KEY (Anonimo) references Anonimo(Ip),
    FOREIGN KEY (Live) references Live(URL)
)

CREATE TABLE SOTTOSCRIVE(
    Utente VARCHAR(30) NOT NULL,
    Canale VARCHAR(30) NOT NULL,
    Data3 DATE NOT NULL,

    FOREIGN KEY (Utente) references Utente(Username),
    FOREIGN KEY (Canale) references Canale(Nome)
)

CREATE TABLE SEGUE(
    Utente VARCHAR(30) NOT NULL,
    Canale VARCHAR(30) NOT NULL,

    FOREIGN KEY (Utente) references Utente(Username),
    FOREIGN KEY (Canale) references Canale(Nome)
)

CREATE TABLE REGISTRA(
    Utente VARCHAR(30) NOT NULL,
    Live VARCHAR(50) NOT NULL,
    Data4 DATE NOT NULL,
    NumSpettatori INT DEFAULT 0,

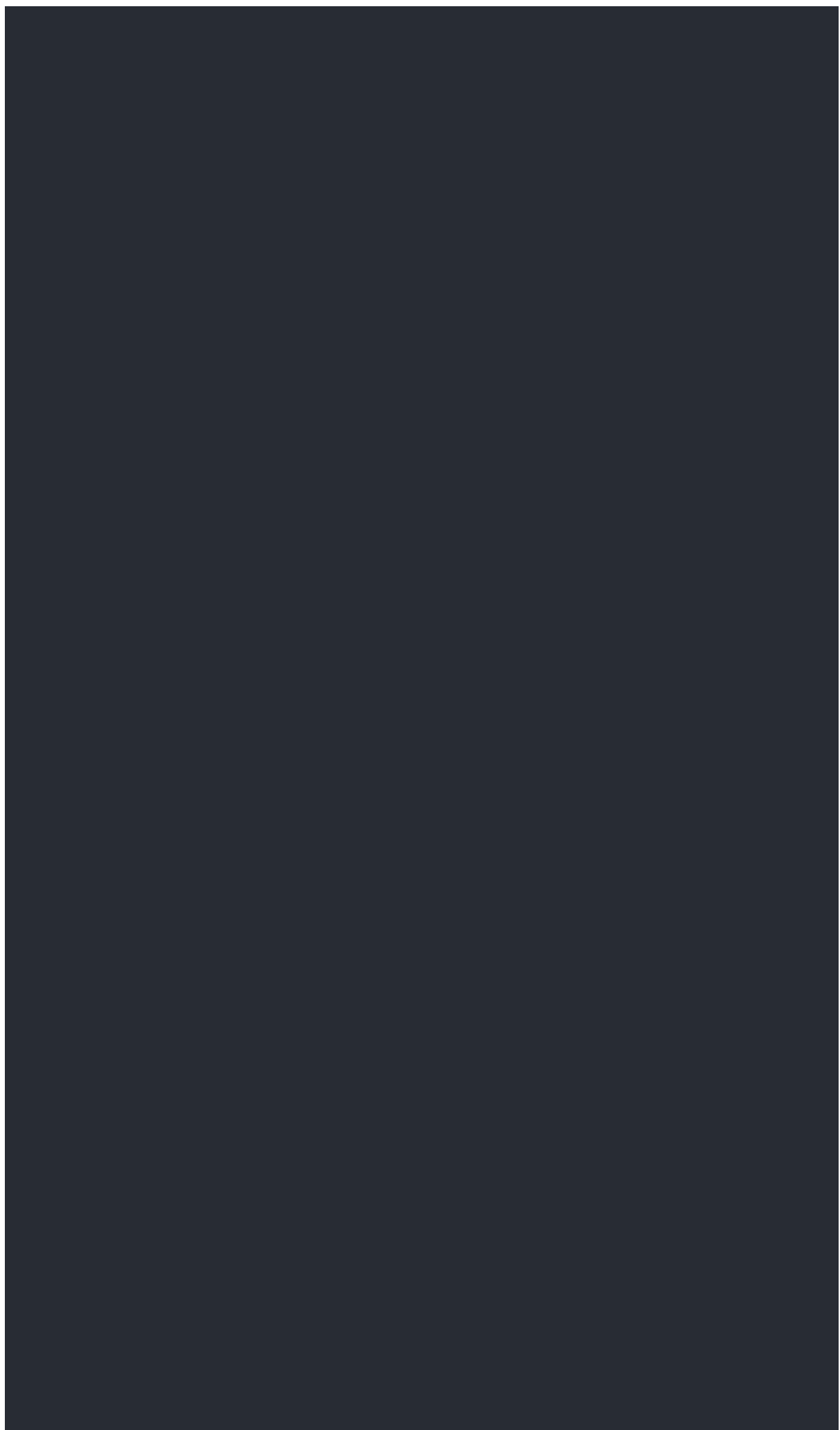
    FOREIGN KEY (Utente) references Utente(Username),
    FOREIGN KEY (Live) references Live(URL)
)

CREATE TABLE ASSOCIA(
    Tag VARCHAR(20) NOT NULL,
    Live VARCHAR(50) NOT NULL,

```



```
    FOREIGN KEY (Tag) references Tag(Nome),  
    FOREIGN KEY (Live) references Live(URL)  
)  
  
CREATE TABLE AVVIA(  
    Utente VARCHAR(30) NOT NULL,  
    Live VARCHAR(30) NOT NULL,  
    Data5 DATE NOT NULL,  
  
    FOREIGN KEY (Utente) references Utente(Username),  
    FOREIGN KEY (Live) references Live(URL)  
)
```



3.2. DML di popolamento di tutte le tabelle di database

Inserire una categoria

```
INSERT INTO CATEGORIA (Nome) VALUES ('Sport');  
INSERT INTO CATEGORIA (Nome) VALUES ('Musica');
```

Inserire un utente

```
INSERT INTO UTENTE (Username, Email, Pass, DataN, Telefono, Affiliato,  
NumeroMin, NumeroMedSpett, Portafoglio) VALUES  
( 'user1', 'user1@example.com', 'password1', '1990-01-15', 1234567890, 1,  
10, 20, 1),  
( 'user2', 'user2@example.com', 'password2', '1985-05-20', 9876543210, 0,  
5, 15, 2)
```

Inserire un utente anonimo

```
INSERT INTO ANONIMO (Ip) VALUES  
( '192.168.21.21'),  
( '192.168.22.22');
```

Inserire un portafoglio

```
INSERT INTO PORTAFOGLIO (Indice, Saldo) VALUES  
(1, 1000.00),  
(2, 500.00);
```

Inserire un tag

```
INSERT INTO TAG (Nome) VALUES  
( 'Tag1'),  
( 'Tag2'),  
( 'Tag3');
```

Inserire una live

```
INSERT INTO LIVE (Indirizzo, Calendario, Titolo, NumSpettatori,  
Terminata, Categoria) VALUES  
( 'Address1', 'Calendario1', 'Live1', 50, FALSE, 'Categoria1'),  
( 'Address2', 'Calendario2', 'Live2', 30, TRUE, 'Categoria2'),  
( 'Address3', 'Calendario3', 'Live3', 70, FALSE, 'Categoria3');
```

Inserire una clip

```
INSERT INTO CLIP (Titolo, Durata, NumeroVisite, Video, Canale) VALUES  
( 'Clip1', '00:05:00', 50, 'Video1', 'Canale1'),  
( 'Clip2', '00:03:30', 25, 'Video2', 'Canale2'),  
( 'Clip3', '00:07:45', 80, 'Video3', 'Canale3'),
```

Inserire un canale

```
INSERT INTO CANALE (Nome, Calendario, Immagine, Social, Descrizione,
Trailer, numFollow) VALUES
('Canale1', 'Calendario1', 'image1.jpg', 'social1.com', 'Description1',
'trailer1.mp4', 120),
('Canale2', 'Calendario2', 'image2.jpg', 'social2.com', 'Description2',
'trailer2.mp4', 80),
('Canale3', 'Calendario3', 'image3.jpg', 'social3.com', 'Description3',
'trailer3.mp4', 150);
```

Inserire un video

```
INSERT INTO VIDEO (Titolo, Durata, NumeroVisite, Live) VALUES
('Video1', '01:30:00', 500, 'Address1'),
('Video2', '00:45:00', 300, 'Address2'),
('Video3', '02:15:00', 700, 'Address3');
```

Inserire un calendario

```
INSERT INTO CALENDARIO (Nome, Orario, Datac) VALUES
('Calendario1', '12:00:00', '2023-01-01'),
('Calendario2', '15:30:00', '2023-02-15'),
('Calendario3', '18:45:00', '2023-03-22');
```

Inserire un nuovo messaggio tra utenti

```
INSERT INTO SCRIVE1 (Utente1, Utente2, Data1) VALUES
('user1', 'user2', '2023-01-05'),
('user2', 'user3', '2023-02-10');
```

Inserire un nuovo messaggio nella live

```
INSERT INTO SCRIVE2 (Utente, Live, Tempo) VALUES
('user1', 'Address1', '12:00:00'),
('user2', 'Address2', '13:30:00'),
('user3', 'Address3', '14:45:00'),
('user4', 'Address4', '16:15:00'),
('user5', 'Address5', '18:00:00');
```

Inserire donazione

```
INSERT INTO DONA (Utente1, Utente2, Data1, valore) VALUES
('user1', 'user2', '2023-01-05', 50.00),
('user2', 'user3', '2023-02-10', 30.00);
```

Inserire che utente visiona live

```
INSERT INTO VISIONA1 (Utente, Live) VALUES
('user1', 'Address1'),
('user2', 'Address2');
```

Inserire che utente anonimo visiona live

```
INSERT INTO VISIONA2 (Anonimo, Live) VALUES
('192.168.1.1', 'Address1');
```

```
('192.168.2.2', 'Address2');
```

Inserire che utente si abbona al canale

```
INSERT INTO SOTTOSCRIVE (Utente, Canale, Data3) VALUES  
( 'user1', 'Canale1', '2023-01-05'),  
( 'user2', 'Canale2', '2023-02-10');
```

Inserire che utente avvia live

```
INSERT INTO AVVIA (Utente, Live, Data5) VALUES  
( 'user1', 'Address1', '2023-01-01'),  
( 'user2', 'Address2', '2023-02-02');
```

Inserire registrazione di numero spettatori medi della live

```
INSERT INTO REGISTRA (Utente, Live, Data4, NumSpettatori) VALUES  
( 'user1', 'Address1', '2023-01-01', 100),  
( 'user1', 'Address1', '2023-01-02', 120),  
( 'user1', 'Address1', '2023-01-03', 90),  
( 'user1', 'Address1', '2023-01-04', 110),  
( 'user1', 'Address1', '2023-01-05', 80),  
( 'user1', 'Address1', '2023-01-06', 95),  
( 'user1', 'Address1', '2023-01-07', 105),  
( 'user1', 'Address1', '2023-01-08', 115),  
( 'user1', 'Address1', '2023-01-09', 130);
```

Inserire associazione tra live e tag

```
INSERT INTO ASSOCIA (Tag, Live) VALUES  
( 'Tag1', 'Address1'),  
( 'Tag2', 'Address2'),  
( 'Tag3', 'Address3');
```

Inserire collegamento tra utente e canale

```
INSERT INTO SEGUE (Utente, Canale) VALUES  
( 'user1', 'Channel1'),  
( 'user2', 'Channel2'),  
( 'user3', 'Channel3'),  
( 'user4', 'Channel4'),  
( 'user5', 'Channel5');
```

3.3. Operazione di Cancellazione e modifica

```
DELETE FROM `utente`  
WHERE Username = "user1";
```

```
DELETE FROM `canale`  
WHERE Nome = "canale1";
```

```
UPDATE `utente`  
SET Pass = "nuova";  
WHERE Email = "user1@example.com";
```

```
UPDATE `canale`  
SET Calendario = "Calendario2";  
WHERE nome = "Canale1";
```