

Progetto di Basi di Dati

Servizio di Live Streaming

Francesco Mauro, Riccardo Oro

2023

Contents

Chapter 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 alecattelan. (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.

Per ogni creatore di contenuti, si memorizzano il numero di live effettuate, il numero di minuti trasmessi, il numero medio di spettatori simultanei. Inoltre, sulla pagina del canale viene visualizzato il numero di follower.

Quando uno streamer rispetta determinati parametri di performance (un minimo di 500 minuti trasmessi, una media di tre o più spettatori simultanei, almeno 50 follower), può diventare *affiliate*.

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

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

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

La base di dati deve supportare le seguenti operazioni:

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

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

Chapter 2

Progettazione Concettuale

2.1 Glossario

Termine	Descrizione	Sinonimi	Collegamento
Utente	Chi usufruisce del servizio	Spettatore, Streamer	Spettatore, Streamer, Registrato, Anonimo
Spettatore	Colui che guarda le live	Utente	Registrato, Anonimo
Streamer	Colui che gestisce un canale e intrattiene gli spettatori		Utente, Registrato, Canale
Registrato	Utente che ha fornito nome utente, password, data di nascita e numero di telefono o indirizzo email		Utente
Live Streaming	Trasmissione in diretta che permette di interagire con il pubblico in tempo reale		Canale, Streamer
Canale	Ogni streamer ha un canale dove può caricare i suoi contenuti e eventuali informazioni esterne		Streamer, Live Streaming

2.2 Revisione dei requisiti

2.2.1 Fase relativa agli utenti

- Gli utenti si dividono in registrati o anonimi

2.2.2 Fase relativa a utenti anonimi

- Gli utenti anonimi possono visitare i canali senza doversi per forza registrare, ma non possono interagire

2.2.3 Fase relativa a utenti registrati

- Gli utenti per registrarsi devono:
 - Registrarsi fornendo: username, password, email o numero di telefono, data di nascita

E possono accedere a:

- Canale
- chat privata o pubblica
- portafoglio (per eventuali donazioni)
- possono diventare *supporter* tramite *subscription*

Per utenti che creano contenuti invece:

- Il numero di live effettuate
- il numero di minuti trasmessi
- il numero medio di spettatori ¹

2.2.4 Fase relativa al canale

Un canale è composto da:

- Descrizione
- Lista dei social
- Immagine del profilo
- Trailer
- Live
- Video e clip ²
- Ore di streaming

¹Tutti questi dati si possono usare per far diventare uno *streamer affiliate*

²Queste non sono in streaming ma vengono salvate

2.2.5 Fase relativa alle live

- Possono essere viste da tutti gli utenti
- Iniziano a un determinato orario Viene memorizzato:
 - Il numero medio di spettatori
 - Chat
 - il titolo
- A ogni live corrisponde un URL

Fase relativa a una chat

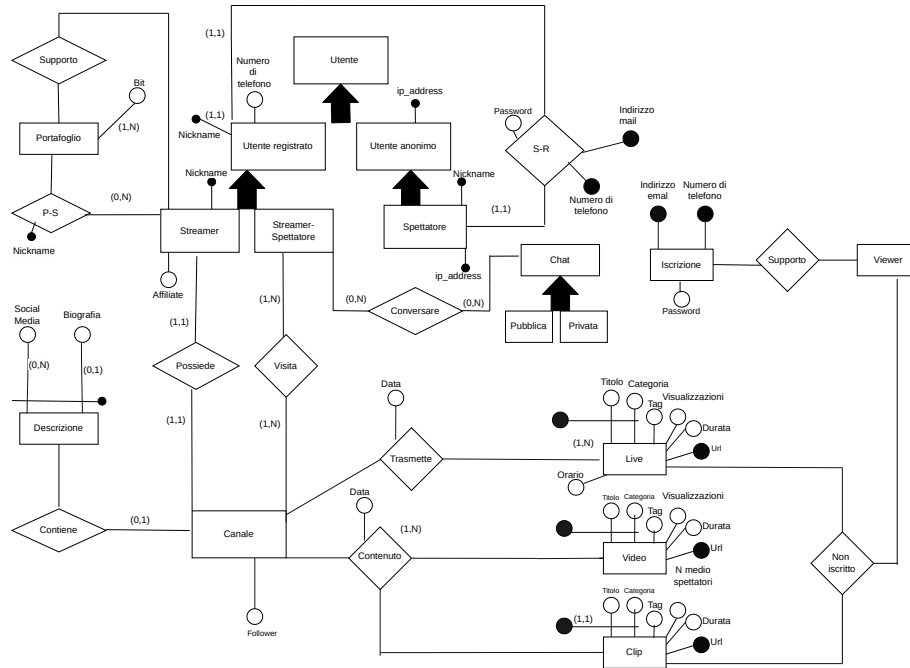
- Solo gli utenti registrati possono accedere alla chat Può essere:
 - Privata (Tra utente e utente)
 - Pubblica

2.2.6 Fase relativa alle Operazioni

- Registrazione utente
- donazioni ³
- Diventare follower di un canale
- Avviare una chat privata tra utenti registrati

³A utenti registrati

2.3 Schema E-R + business rules



2.3.1 Vincoli di integrità

- I video sono degli estratti di live passate.
- Le clip sono dei video di breve durata.
- I bit può essere acquistato nella piattaforma.
- Un utente per registrarsi sulla piattaforma deve fornire indirizzo email o numero di telefono.
 - Un utente anonimo può guardare la live trasmessa dallo streamer ma non può interagire con esso.
- All'interno del canale sono presenti live passate ma anche la trasmissione in tempo reale di una eventuale live in atto.
- Ogni contenuto all'interno della piattaforma è identificato con un indirizzo **Url**.
- Se uno *streamer* rispetta i *parametri di performance* può diventare *affiliate*.
- Il portafoglio è costituito da *bit*, con i quali il **follower** può supportare lo streamer
- Il viewer non può supportare lo streamer.
- La chat permette agli utenti di comunicare l'uno con l'altro sia pubblicamente sia privatamente.
- Il nome del canale è lo stesso nome che l'utente inserisce durante la registrazione alla piattaforma.
- All'interno del canale è presente una biografia.
- All'interno del canale è presente una lista di link ad altri social media.
- Le live vengono programmate attraverso un calendario
- Ogni portafoglio è indentificato con il nickname dell'utente registrato.
- Ogni utente può donare un numero \geq di 0 bit ad uno streamer.
- Il nome dello streamer è lo stesso che viene inserito durante la fase di registrazione alla piattaforma.
- Il nome dello streamer-spettatore è lo stesso che viene inserito durante la fase di registrazione alla piattaforma.
- Se si vuole inviare un messaggio pubblico bisogna modificare il valore booleano presente nella tabella in TRUE e bisogna inserire NULL al campo `nick_ricevente` nella tabella chat.

Chapter 3

Progettazione Logica

3.1 Tavola dei volumi

Concetto	Tipo	Volume	Motivazione
Spettatore	E	1.000.000	Si ipotizza un totale di 1.000.000 spettatori
Utenti Registrato	E	950.000	Si ipotizza che la maggior parte degli utenti sia registrata
Utenti Anonimi	E	50.000	Una minima parte degli spettatori non è Registrato
Streamer-Spettatore	E	100.000	Si ipotizza che 100.000 spettatori siano anche streamer
Streamer	E	10.0000	Utenti che decidono di svolgere attività streaming sulla piattaforma
Descrizione	E	10.000	Il numero di descrizioni è pari al numero di descrizioni
Canali	E	10.000	Utenti registrati alla piattaforma e che hanno un canale attivo
Video	E	100.000	Contenuto video caricato sul canale dello streamer
Live	E	1.000.000	Contenuto video live in onda sul canale in tempo reale o live passata
Clip	E	100.000	Estratto di breve durata del contenuto in live
Portafoglio	E	1000000	Quantitativo di moneta virtuale "bit" acquistabile sulla piattaforma
Iscrizione	E	750.000	Supponiamo che 750.000 spettatori si iscrivano almeno a un canale
Chat	E	1.000.000.000	Supponiamo che vengano mandati 1.000.000.000 di messaggi tra privati e pubblici
Conversare	R	1.000.000.000	Quantitativo di messaggi scambiati su chat pubbliche o private tra utenti
Contenuto	R	1.200.000	Il numero di contenuti è pari alla somma dei volumi di live video e clip
Visita	R	1.200.000	Il numero di visite è pari alla somma dei volumi di live video e clip
Portafoglio-Streamer	R	1.000.000	Relazione tra Portafoglio e streamer
Streamer-Utente registrato	R	1.000.000	Relazione utente Spettatore non registrato si registra alla piattaforma
Non iscritto	R	1.000.000.000	Utenti non registrati sulla piattaforma
Possiede	R	10.000	Quantitativo di canali complessivi presenti sulla piattaforma
Contiente	R	1.200.000	Quantitativo di contenuti video presenti sulla piattaforma
Supporto	R	1.300.000	Quantitativo di supporto monetario ricevuto dallo streamer

3.2 Operazioni Previste

1. Registrazione:
2. Visualizzazione Chat:
3. Visualizzazione Live:
4. Visualizzazione Video:
5. Visualizzazione Clip:
6. Visualizzazione Chat:
7. Visualizzazione Descrizione:
8. Donazione:
9. Visualizzazione Canale
10. Calcolo Numero Spettatori Medi:
11. Calcolo Donazioni Ricevute:
12. Calcolo Numero di live effettuate
13. Calcolo Numero di video effettuate
14. Calcolo Numero di clip effettuate

3.2.1 Tavola delle operazioni

Operazioni	Tipo	Frequenza	Motivazione
Registrazione	I	3.000/giorno	un utente può registrarsi alla piattaforma inserendo i propri dati personali
Visualizzazione Clip	I	50.000/giorno	un utente può visualizzare una clip
Visualizzazione Live	I	30.000/giorno	un utente può visualizzare una live
Visualizzazione Video	I	30.000/giorno	un utente può visualizzare un video
Visualizzazioine Chat	I	1.000.000/giorno	un utente può visualizzare la chat

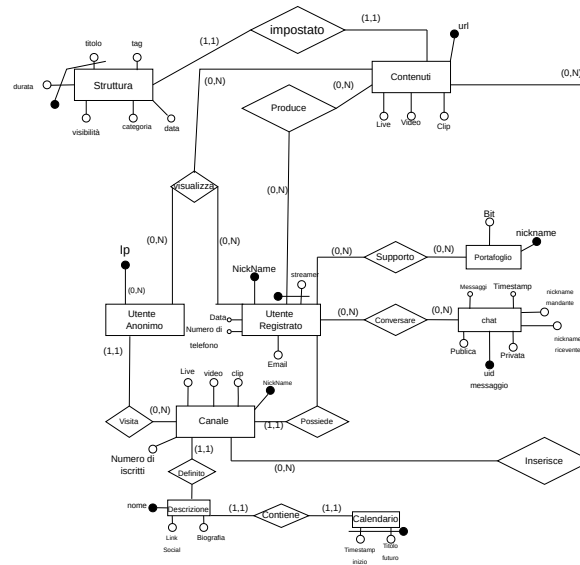
Continua alla pagina successiva

	Operazioni	Tipo	Frequenza	Motivazione	
	Donazione		I	10.000/giorno	un utente può donare ad uno streamer
	Caricamento di un contenuto all'interno del canale		I	5.000/settimana	Un utente può decidere di caricare sul proprio canale contenuti di tipo video ,live ,clip.
	Acquisto moneta bit da utente		I	1.000.000/settimana	un utente può acquistare bit .
	Calcolo Numero Spettatori Medi		I	1/giorno	la piattaforma calcola il numero di spettatori medi
	Calcolo Numero di Live effettuate		I	1/giorno	la piattaforma calcola il numero di live effettuate da uno streamer
	Calcolo Numero di Clip effettuate		I	1/giorno	la piattaforma calcola il video di live effettuate da uno streamer
	Calcolo Numero di Video effettuate		I	1/giorno	la piattaforma calcola il clip di live effettuate da uno streamer
	Calcolo Numero di contenuti presenti sul canale		B	1/giorno	La piattaforma stampa il numero di contenuti presenti per ogni canale.
	Controllo condizioni affiliate		B	1/giorno	la piattaforma controlla se uno streamer rispetta le condizioni per diventare affiliate
	Controllo classifica streamer più seguiti		B	1/settimana	la piattaforma controlla se uno streamer rispetta le condizioni per diventare affiliate.

Continua alla pagina successiva

Operazioni	Tipo	Frequenza	Motivazione
Calcolo numero iscrizioni al canale	B	1/giorno	La piattaforma calcola quanti utente hanno deciso di seguire il canale dello streamer.
Calcolo numero bit acquistati dagli utenti	B	1/giorno	La piattaforma calcola il totale di bit acquistati dagli utenti e inseriti nel Portafogli.

3.3 Ristrutturazione schema E-R



3.4 Analisi delle Ridondanze

Le ridondanze trovate sono le seguenti:

1. La relazione *visualizza* collegata all'entità *utente anonimo* è ridondante in quanto l'informazione è ricavabile dall'entità *Canale*
2. Il nome del canale è derivabile dal nickname dell'utente registrato
3. Gli attributi dell'entità *Contenuti* sono ricavabili dall'entità *Canale*
4. I messaggi inviati dagli utenti non hanno un identificativo che indica se è un messaggio pubblico o privato ,quindi se un messaggio si trova all'interno di una chat pubblica o privata esso sarà comunque un messaggio quindi è ricavabile dall'entità chat.

Ridondanza dei messaggi pubblici/privati

- È possibile analizzare che in presenza di ridondanza il quantitativo di memoria contenuto nell'entità chat di messaggi è nettamente superiore ,poichè l'attributo Chat immagazina qualunque tipo di messaggi senza badare al fatto che sia pubblica o privata , il fatto che la chat sia pubblica o privata è influente per il messaggio ,poichè rimane comunque un messaggio. Senza la ridondanza i messaggi vengono ri identificati e gestiti dalle singole chat in modo che ogni chat ha una sua gestione e quindi ha una quantità diversa di messaggi da elaborare,e quindi essi non vengono raggruppati in un unica sezione di memoria.

Si ipotizzano 1.000.000.000 di messaggi scambiati, con la ridondanza un messaggio non viene distinto tra pubblico e privato, quindi viene contato doppio

Analisi con ridondanza

Tempo impiegato : $\frac{2.000.000.000 \text{messaggi} * 1 \text{ms}}{\text{messaggi} * 2} \simeq 32$ minuti per salvare tutti i messaggi.

Tempo impiegato: Spazio:Si ipotizza 100 kb a messaggio.

Concetto	Tipo	Accessi	Tipo
Utente Registrato	R/W	Conversare	1.000.000.000

¹i messaggi sono il doppio di quelli effettivi in quanto non vengono distinti, perazioni di read and write

Analisi senza ridondanza

Tempo impiegato : $\frac{1.000.000.000 * 0.5ms}{messaggi * 2}^2 \simeq 16$ minuti per salvare tutti
Spazio: Si ipotizza 100 kb a messaggio.

Concetto	Tipo	Accessi	Tipo
Utente Registrato	R/W	Conversare	2.000.000.000

3.5 Eliminazioni delle Generalizzazioni

Le generalizzazioni presenti nello schema E-R sono:

1. **Utente:** È una generalizzazione totale tra le entità *Utente registrato* e *Utente anonimo*, poichè o si è registrati alla piattaforma o si è anonimi
2. **Streamer e Streamer-Spettatore:** È una generalizzazione parziale tra le entità *Streamer* e *Streamer-Spettatore*, in quanto l'utente registrato si ritrova a svolgere il ruolo o di streamer o di spettatore, ma sempre essendo uno streamer
3. **Spettatore - Utente Anonimo:** È una generalizzazione sovrapposta, poichè l'utente anonimo può essere solo spettatore'
4. **Chat:** È una generalizzazione totale tra le entità *Chat privata* e *Chat pubblica*. Perché l'entità chat si divide in due sottoinsiemi, ovvero chat pubblica e chat privata.

3.6 Partizionamento/Accorpamento di entità e associazioni

Si è deciso di accorpare l'entità chat, così indica in maniera più coincisa che un utente registrato può comunicare in maniera pubblica o privata''

3.7 Business rules dell'E-R ristrutturato

- Gli attributi live ,video ,clip presenti nell'entità **canale** indicano un oggetto finito e caricato al suo interno.
- Gli attributi live ,vide ,clip presenti sull'entità **contenuti** indicano un oggetto che ancora o è in fase di produzione o è un prodotto finito ma non caricato sul canale del proprietario.
- Gli utenti anonimi non possono supportare gli streamer.

²operazioni di read and write

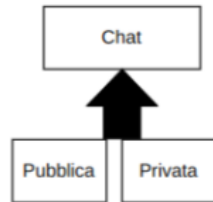


Figure 3.1: Entità chat dopo l'accorpamento

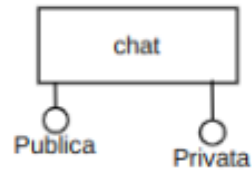


Figure 3.2: Entità chat prima dell'accorpamento

- Gli streamer sono utenti registrati che caricano o trasmettono contenuti.
- Ogni utente registrato ha un nickname.
- Il nickname scelto dall'utente registrato sarà anche il nome del canale.
- Ogni canale ha una Descrizione.
- Le live sono contenuti o in tempo reale.
- I video sono live già concluse.
- Le clip sono brevi estratti di video.
- Un utente anonimo non può conversare con altri utenti.
- Un utente anonimo non possiede un canale.

3.8 Schema Relazionale

- Utente registrato(nickname,email,password,affiliato,numero di telefono)
- Utente anonimo(ip address).
- Canale(live, video , clip,n_iscritti,nickname,iscritti).
- Descrizione(Nome,link_social,biografia)
- Calendario(timestamp_inizio,titolo_futuro)
- Contenuti(Url,live ,video , clip).
- Portafoglio(bit,nickname).
- Chat(UID messaggio,pubblica ,privata, nickname mandante ,nickname ricevente).
- struttura(visibilità , data, categoria , titolo ,durata, tag)
- Visita(utente anonimo ,canale).
- Possiede(utente registrato,canale).
- Inserisce(canale,contenuti).
- Produce(utente registrato,contenuti).
- Supporto(utente registrato ,portafoglio).
- Conversare(utente registrato ,chat).
- Impostato(struttura ,contenuti)

Chapter 4

DDL di creazione del DB

```

1 CREATE TYPE contents AS ENUM ('live', 'video', 'clip');
2 CREATE TABLE IF NOT EXISTS Canale
3 (
4     nickname varchar(20) PRIMARY KEY UNIQUE,
5     FOREIGN KEY (nickname) REFERENCES Utente_registrato(nickname),
6     iscritti integer DEFAULT 0,
7     social_media_list varchar(45),
8     -- FOREIGN KEY ( social_media_list) REFERENCES descrizione(social),
9     --Dato deribale da tabella descrione possibile ridontanda bhoooo
10     --?!?!?!
11     biografia varchar(280)
12 );

```

```

1 --@block initialize Utente registrato
2 CREATE TABLE IF NOT EXISTS Utente_registrato
3 (
4     nickname varchar (45) PRIMARY KEY UNIQUE,
5     streamer BOOLEAN DEFAULT FALSE,
6     email varchar(45),
7     password varchar(13) NOT NULL,
8     DataRegistrazione DATE NOT NULL DEFAULT CURRENT_DATE,
9     NumeroDiTelefono char (10),
10    affiliate BOOLEAN DEFAULT FALSE,
11    Constraint user_valid_email CHECK (
12        email ~* '^[A-Za-z0-9._+%~]+@[A-Za-z0-9.-]+[.][A-Za-z]+$'
13    )
14 )
15
16 );

```

```

1 --@block initalize Utente anonimo
2 CREATE TABLE IF NOT EXISTS Utente_anonimo
3 (
4     ip_address varchar(32),
5     check_ip integer DEFAULT 0 NOT NULL,
6     CONSTRAINT check_ip_range CHECK (check_ip >= 0 AND check_ip <= 255),
7     PRIMARY KEY (ip_address),
8     CONSTRAINT user_valid_ip CHECK (
9         ip_address ~
10         '^(\\d|[1-9]\\d|1\\d\\d|2([0-4]\\d|5[0-5]))\\.\\d|[1-9]\\d|1\\d\\d|2([0-4]\\d|5[0-5]))\\.\\d|[1-9]\\d|1\\d\\d|2([0-4]\\d|5[0-5]))'
11     )
12 );

```

```

1 CREATE TABLE IF NOT EXISTS Descrizione (
2     nome varchar(20),
3     social text default NULL,
4     biografia text default NULL,
5     FOREIGN KEY (nome) REFERENCES Canale(nickname) ON DELETE CASCADE,
6     constraint link_valid check (
7         social ~
8         '^(http|https)://(www.)?[a-zA-Z0-9@:%.\+~#?&//=]{2,256}\.[a-z]{2,6}\b([-a-zA-Z0-9@:%.\+~#?&//=]*)$',
9     );
10 );
11
12
13 -- TODO solve regex error

```

```

1 --@block initialize Chat_privata
2 CREATE EXTENSION IF NOT EXISTS "uuid-oss";
3 CREATE TABLE IF NOT EXISTS chat
4 (
5     id_messaggio uuid DEFAULT uuid_generate_v4 (),
6     pubblica BOOLEAN DEFAULT FALSE,
7     messaggio varchar(100) DEFAULT NULL,
8     nick_mandante VARCHAR(45) ,
9     nick_ricevente VARCHAR(45),
10    tempo TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
11    PRIMARY KEY ( id_messaggio)
12 );

```

```

1 --@block initialize Video
2 CREATE TABLE Video
3 (
4     titolo varchar(90) default NOT NULL,
5     durata decimal (6,4) default 0,
6     categoria integer default NULL,
7     num_spettatori_medio float default 0,
8     tag varchar(20) default NULL,
9     constraint check_tag check (
10         tag ~ '[A-Za-z0-9]'
11     )
12     url varchar(45),
13     constraint check_url check (
14         url ~ '^(http|https)://(www.)?
15 + [a-zA-Z0-9@:%.\+~#?&//=]{2,256}\.[a-z]
16 + {2,6}\b([-a-zA-Z0-9@:%.\+~#?&//=]*)'
17     )
18     --TODO : da rivedere
19     -- Foreign Key (titolo) REFERENCES ()
20 );
21 CREATE TABLE clip
22 (
23     titolo varchar(45) default NOT NULL ,
24     create_domain durata as decimal(6,4) default NOT NULL
25     check(value >= 0.1 AND value <= 0.60),
26
27 );

```

4.1 DML di popolamento del DB

```

1 INSERT INTO utente_registrato(nickname,email,password,DataRegistrazione,
2 NumeroDiTelefono,affiliate)
3 VALUES (
4     'riccardo',
5     'riccardo.oro@edu.unito.it',

```



```

6      'password',
7      '2023-01-01',
8      '3791608787',
9      true
10 );
11
12 INSERT INTO utente_registrato(nickname,email,password,DataRegistrazione,
13 NumeroDiTelefono,affiliate)
14 VALUES (
15     'francesco',
16     'francesco.mauro590@edu.unito.it',
17     'password5',
18     '2021-07-01',
19     '3423292372',
20     false
21 );
22
23 INSERT INTO utente_registrato(nickname,email,password,DataRegistrazione,
24 NumeroDiTelefono,affiliate)
25 VALUES (
26     'santiago',
27     NULL,
28     'storia',
29     '2023-09-25',
30     '3286068918',
31     true
32 );
33
34 INSERT INTO utente_registrato(nickname,email,password,DataRegistrazione,
35 NumeroDiTelefono,affiliate)
36 VALUES (
37     'Tumblurr',
38     'Tumblurr@blur.com',
39     'lugoizzi',
40     '2006-12-21',
41     '3911822225',
42     true
43 );
44
45 INSERT INTO utente_registrato(nickname,email,password,DataRegistrazione,
46 NumeroDiTelefono,affiliate)
47 VALUES (
48     'giovanna',
49     'giovannaLabionda@parruchieria.org',
50     'password4',
51     '2023-05-06',
52     '3911855685',
53     true
54 );
55
56 INSERT INTO utente_registrato(nickname,email,password,DataRegistrazione,
57 NumeroDiTelefono,affiliate)
58 VALUES (
59     'sarah',
60     'sarah.connor@skynet.net',
61     't-800',
62     '2023-01-22',
63     NULL,
64     true
65 );
66
67 INSERT INTO utente_registrato(nickname,email,password,DataRegistrazione,
68 NumeroDiTelefono,affiliate)
69 VALUES (
70     'truceklan',
71     NULL,
72     'ministero',
73     '2022-12-13',

```

```

74     '3426661312',
75     true
76 );
77
78 INSERT INTO utente_registrato(nickname,email,password,DataRegistrazione,
79 NumeroDiTelefono,affiliate)
80 VALUES (
81     'TheRealMarzaa',
82     'therealmarzaa@gmail.com',
83     'password78',
84     '2023-12-21',
85     '3472213882',
86     false
87 );
88
89 INSERT INTO utente_registrato(nickname,email,password,DataRegistrazione,
90 NumeroDiTelefono,affiliate)
91 VALUES (
92     'bassi maestro',
93     'northofloreto@outlook.com',
94     'fotodigruppo',
95     '2023-01-01',
96     '3456972301',
97     true
98 );
99
100 INSERT INTO utente_registrato(nickname,email,password,DataRegistrazione,
101 NumeroDiTelefono,affiliate)
102 VALUES (
103     'piero',
104     'piero@gmail.com',
105     'password',
106     '2023-01-01',
107     NULL,
108     false
109 );
110
111 INSERT INTO utente_registrato(nickname,email,password,DataRegistrazione,
112 NumeroDiTelefono,affiliate)
113 VALUES (
114     'GiorgioneOrtoeCucina',
115     'giorgioneortoecucina@test.com',
116     'password3',
117     '2023-01-01',
118     NULL,
119     false
120 );

```



```

1 INSERT INTO canale ( nickname , iscritti , social_media_list ,biografia)
2 VALUES(
3     'Tumblurr',
4     1300000,
5     'https://www.instagram.com/blurs7/',
6     'Gaming e altro'
7 );
8 INSERT INTO canale ( nickname , iscritti , social_media_list ,biografia)
9 VALUES(
10     'TheRealMarzaa',
11     1400000,
12     'https://www.instagram.com/therealmarzaa/',
13     'gameplay'
14 );
15 INSERT INTO canale ( nickname , iscritti , social_media_list ,biografia)
16 VALUES(
17     'riccardo',
18     1000,
19     'www.instagram/non_lo_so_rick',
20     'Faccio qualche partita a qualcosa e parlo di cose '

```

```

21 );
22 INSERT INTO canale ( nickname , iscritti , social_media_list ,biografia)
23 VALUES(
24     'francesco',
25     0,
26     'www.instagram/fra01',
27     '../'
28 );
29
30 INSERT INTO canale(nickname,iscritti,social_media_list,biografia)
31 VALUES(
32     'santiago',
33     69000,
34     'www.twitter.com/cuadrosenelmare',
35     'Ore e ore passate a parlare del passato (ovvero di storia).'
36     '-----'
37 );
38
39
40 INSERT INTO canale ( nickname , iscritti , social_media_list ,biografia)
41 VALUES(
42     'GiorgioneOrtoeCucina'
43     1.000.000,
44     'www.instagram/GiorgioneOrtoeCucina',
45     'Cucino prelibatezze'
46 );
47
48
49 --DA TESTARE DOPO MODIFICA TABELLA CANALE

```

```

1  INSERT INTO descrizione(nome, social, biografia)
2  VALUES (
3      'Riccardo Oro',
4      'www.instagram.com/Riccardo_Oro',
5      'Live streaming su come risolvere segmentation fault'
6  );
7
8  INSERT INTO descrizione(nome, social, biografia)
9  VALUES (
10     'Francesco Mauro',
11     'www.instagram.com/Francesco_Mauro',
12     NULL
13 );
14
15 INSERT INTO descrizione(nome, social, biografia)
16 VALUES (
17     'Santi',
18     'www.facebook.com/Santi',
19     'Ore e ore passate a parlare del passato (ovvero di storia).'
20 );
21
22 INSERT INTO descrizione(nome, social, biografia)
23 VALUES (
24     'Sarah',
25     'www.blogofskynet.org',
26     'Come le AI domineranno il mondo'
27 );
28
29 INSERT INTO descrizione(nome, social, biografia)
30 VALUES (
31     'Macco',
32     NULL,
33     'C# come unica religione'
34 );
35
36 INSERT INTO descrizione(nome, social, biografia)
37 VALUES (
38     'Alessandra',

```

```

39 'www.sociale.com/ale1312',
40 'Perch comprare un carlino?'
41 );
42
43 -- running this give this error: ERROR: invalid regular expression: invalid
    repetition count(s) SQL state: 2201B

```

```

1 INSERT INTO descrizione(nome, social, biografia)
2 VALUES (
3     'Riccardo Oro',
4     'www.instagram.com/Riccardo_Oro',
5     'Live streaming su come risolvere segmentation fault'
6 );
7
8 INSERT INTO descrizione(nome, social, biografia)
9 VALUES (
10    'Francesco Mauro',
11    'www.instagram.com/Francesco_Mauro',
12    NULL
13 );
14
15 INSERT INTO descrizione(nome, social, biografia)
16 VALUES (
17    'Santi',
18    'www.facebook.com/Santi',
19    'Ore e ore passate a parlare del passato (ovvero di storia).'
20 );
21
22 INSERT INTO descrizione(nome, social, biografia)
23 VALUES (
24    'Sarah',
25    'www.blogofskynet.org',
26    'Come le AI domineranno il mondo'
27 );
28
29 INSERT INTO descrizione(nome, social, biografia)
30 VALUES (
31    'Macco',
32    NULL,
33    'C# come unica religione'
34 );
35
36 INSERT INTO descrizione(nome, social, biografia)
37 VALUES (
38    'Alessandra',
39    'www.sociale.com/ale1312',
40    'Perch comprare un carlino?'
41 );
42
43 -- running this give this error: ERROR: invalid regular expression: invalid
    repetition count(s) SQL state: 2201B

```

```

1 --alter table chat
2 --add messaggio varchar(100);
3
4 INSERT INTO calendario(titolo_futuro ,inizio)
5 VALUES('LIVE FUTURISTICA','2021-12-20');
6 INSERT INTO chat(messaggio,nick_mandante,nick_ricevente,tempo)
7 VALUES('ciaoquesto unmessaggio','riccardo','francesco',CURRENT_TIMESTAMP);
8 INSERT INTO chat(pubblica,messaggio,nick_mandante,nick_ricevente)
9 VALUES(TRUE,'test1','riccardo',NULL);
10 INSERT INTO chat()

```