



ABSTRACT

Twitch è una piattaforma di live streaming leader del settore. Nato principalmente per lo streaming di videogiochi, negli ultimi anni ha fatto spazio anche a live di qualsiasi tipo (come ad esempio dirette di cucina), contando così un totale di 3 milioni di dirette al mese e 15 milioni di utenti attivi. Per questo motivo si è scelto di aggiornare il database della piattaforma.

Su Twitch un dipendente/streamer (stato che gli viene assegnato quando raggiungono dei requisiti minimi di fama) rende disponibili le sue dirette sul suo canale (specificando la categoria in cui esse vengono pubblicate); una volta che queste (dirette) siano finite, rimangono salvate nel canale in modo da poter essere viste pure su richiesta dagli spettatori in un secondo momento; gli spettatori possono visualizzare un video (diretta o registrata) alla volta. Ogni utente può “seguire” più streamer (dipendenti che producono le dirette) così da essere aggiornati sui loro contenuti. Twitch presenta alla propria utenza una classifica giornaliera degli streamer più seguiti. Se un utente segue uno streamer può anche scegliere di abbonarsi per ottenere dei vantaggi esclusivi come l'assenza di pubblicità nel canale a cui si abbona.

Lo spettatore può sostenere uno streamer (anche senza seguirlo) in qualsiasi momento facendo delle donazioni tramite bits (valuta della piattaforma), o direttamente con la carta di credito.

I profitti dello streamer derivano quindi dalle donazioni, abbonamenti, inserzionisti (che presentano le proprie pubblicità all'interno dei video) e da una società alla quale potrebbero essere affiliati.

Twitch infine può organizzare e ospitare pure degli eventi speciali (ad esempio dei tornei) in cui può invitare una serie di partecipanti (streamer) selezionati dalla piattaforma.

ANALISI DEI REQUISITI

Nella base di dati sono presenti i dati degli **Streamers** che hanno una partnership con Twitch; di ogni streamer è noto:

- username
- ore totali di stream (mensili)
- numero di follower
- numero degli abbonati

Lo streamer ha un **Canale** dove pubblica i suoi stream, del canale sono noti:

- data di creazione
- numero di visualizzazioni
- descrizione (opzionale)

Mentre delle **Stream** si sa:

- streamId
- titolo
- data e ora di inizio
- data e ora di fine (opzionale)
- numero di visualizzazioni
- categoria

Una stream può essere visualizzata in **live** (diretta) oppure in un secondo momento **on-Demand** tramite una registrazione della live. Nel primo caso è possibile pure fare delle donazioni allo streamer.

Delle **Donazioni** si deve conoscere:

- username
- data e ora

ed è possibile effettuarli tramite **donazioni in bits** (valuta della piattaforma) o **donazioni con la carta di credito** esprimendo una certa quantità in entrambi i casi.

Della **carta di credito** si vuole memorizzare:

- circuito
- intestatario

- numero (16 cifre)
- data di scadenza
- cvc (3 cifre)

Degli **Utenti** che guardano le stream sono essenziali le informazioni riguardo a:

- username
- nome
- cognome
- email
- password(cifrata)

Oltre agli utenti base ci possono essere pure due categorie che si relazionano con un canale, i **Moderatori** che moderano la chat di una stream e gli utenti **Bannati** da quel canale per vari motivi.

Si devono conoscere i dati principali di un'**Azienda**:

- nome dell'azienda
- partita Iva (11 cifre)
- indirizzo (stato, città, cap, via, civico)

Le aziende possono essere solamente di due tipi, **Inserzionisti** che sponsorizzano un prodotto proprio tramite delle pubblicità nelle stream o **Società** che vengono affiliati con gli streamers e si occupano di trovare le giuste opportunità per quest'ultimi (come ad esempio gli inserzionisti adatti per il target delle stream ecc.)

Ad ogni **Evento** organizzato da Twitch vi possono partecipare vari streamers invitati, si devono memorizzare le informazioni riguardo a:

- nome dell'evento
- edizione
- data evento

Glossario:

Termine	Descrizione	Collegamento
Stream	Principalmente è una diretta video trasmessa sulla piattaforma, può venire anche registrata per essere pubblicato come un video normale in un secondo momento.	Canale, Inserzionisti, Utente
Streamer	Colui che fa le stream, è un dipendente che ha un contratto con la piattaforma	Società, Evento, Canale, Utente
Canale	Spazio in cui sono conservate le stream di ciascun streamer.	Stream, Streamer, Moderatori, Bannati
Utente	Un utente registrato che fa utilizzo della piattaforma per visualizzare le stream.	Streamer, Stream
Follower	Un utente normale che decide di seguire uno streamer (ottiene dei benefici come il ricevimento di una notifica ad ogni nuova stream)	Streamer, Utente
Moderatori	Utenti scelti da uno streamer per moderare nel canale; hanno poteri amministrativi.	Canale Entità figlia di Utente
Bannati	Utenti che non sono più ammessi in un certo canale per un tempo che può essere indefinito	Canale Entità figlia di Utente
Azienda	Organizzazione con partita Iva che ha un collegamento con la piattaforma di Twitch (tramite Streamer o le Stream)	Entità padre di Inserzionisti e Società

Inserzionisti	Organizzazione che pubblicizza un suo prodotto tramite le pubblicità in una stream.	Stream Entità figlia di Azienda
Società	Organizzazione affiliata con alcuni degli dipendenti/streamer per offrire servizi migliori in una stream.	Streamer Entità figlia di Azienda
Live	Diretta video trasmessa in broadcast su un canale.	Donazione Entità figlia di Stream
OnDemand	Una diretta registrata e pubblicata sul canale in modo da renderlo disponibile agli utenti che non hanno partecipato alla diretta.	Entità figlia di Stream
Donazione	Si tratta di un'offerta lasciata da un utente per supportare uno streamer (può avvenire in bits o con una valuta reale).	Live
Donazioni in Bits	Donazioni fatte con una valuta propria di twitch che si può acquistare in precedenza. Le donazioni in bits possono riscattare delle mini ricompense. Sinonimo entità: ConBits	Entità figlia di Donazione
Donazioni con Carta di Credito	Donazioni fatte con una valuta reale e supportate da un pagamento elettronico (tramite carta di credito). Sinonimo entità: ConCreditCard	CreditCard Entità figlia di Donazione
CreditCard	Metodo di pagamento elettronico utilizzato per fare delle donazioni. Ne fanno parte anche le carte di debito e le carte prepagate (es. postepay).	ConCreditCard
Evento	è un evento organizzato personalmente da Twitch in cui vengono invitati vari Streamers con lo scopo di riunire la community della piattaforma.	Streamer

Operazioni Tipiche:

Twitch conta circa 27000 streamer partner.

Controllo che lo streamer sia in live	10000 al giorno
Controllo numero di abbonati	48 volte al giorno
Controllo numero di follower	1440 volte al giorno
Controllo numero di visualizzazioni totali in un canale	144 volte al giorno
Controllo partecipanti a un evento	12 all'anno
Controllo spettatori in una live	10000 al giorno
Controllo numero annunci diversi in una Stream	10000 al giorno
Controllo donazioni	12000 al giorno
Inserimento evento	12 all'anno
Aggiornamento numero di abbonati	2 al giorno
Aggiornamento numero di follower	12 al giorno

Inserimento stream	20000 al giorno
Inserimento donazioni	12000 al giorno

PROGETTAZIONE CONCETTUALE

Lista entità:

Dove non indicato l'attributo è NOT NULL

STREAMER	
Username	varchar(25) primary key
Oretotali	int
numfollower	int
numabbonati	int

UTENTE	
Username	varchar(25) primary key
nome	varchar(20)
cognome	varchar(20)
email	varchar(30)
password	varchar(100)

Utente si specializza in moderatori e bannati

MODERATORI	
DataInizio	date

BANNATI	
---------	--

STREAM	
StreamId	int primary key
dataOrainizio	timestamp
dataOraFine	timestamp
numViews	int
categoria	varchar(20)
titolo	varchar(50)

Stream si specializza in live e on demand

LIVE	
------	--

ON DEMAND

CANALE	
dataCreazione	date
visualizzazioni	int
descrizione	varchar(300) può essere NULL

CREDIT CARD	
<u>numero</u>	bigint primary key
circuito	varchar(20)
intestatario	varchar(20)
dataScadenza	date
cvc	char(3)

AZIENDA		
<u>nomeAzienda</u>		varchar(50) primary key
<u>partitalva</u>		char(11) primary key
Indirizzo	stato	varchar(56)
	città	varchar(85)
	cap	char(5)
	via	varchar(30)
	civico	int

Azienda si specializza in Inserzionisti e Società

INSERZIONISTI

SOCIETÀ

DONAZIONI	
<u>username</u>	varchar(25) primary key
<u>dataOra</u>	datetime primary key

Donazioni si specializza in donazioni in bits e donazioni in soldi

DONAZIONI BITS	
quantitàBits	int

DONAZIONI CREDIT CARD	
quantitàEuro	double

EVENTO: eventi organizzati da Twitch(tornei,concerti)	
<u>nome</u>	varchar(25) primary key
<u>edizione</u>	int primary key
<u>dataEvento</u>	date primary key

Vincoli non rappresentabili tramite schema E-R:

Se un utente viene bannato da un canale, egli non può:

- diventare un follower di quel canale, e di conseguenza abbonarsi.
- non può essere un moderatore di quel canale.
- non può visionare le stream di quel canale.
- non può fare donazioni nelle stream di quel canale..

Nota: tutti i divieti sopra elencati sono validi solo nel periodo in cui l'utente è bannato.

Vincoli di derivazione:

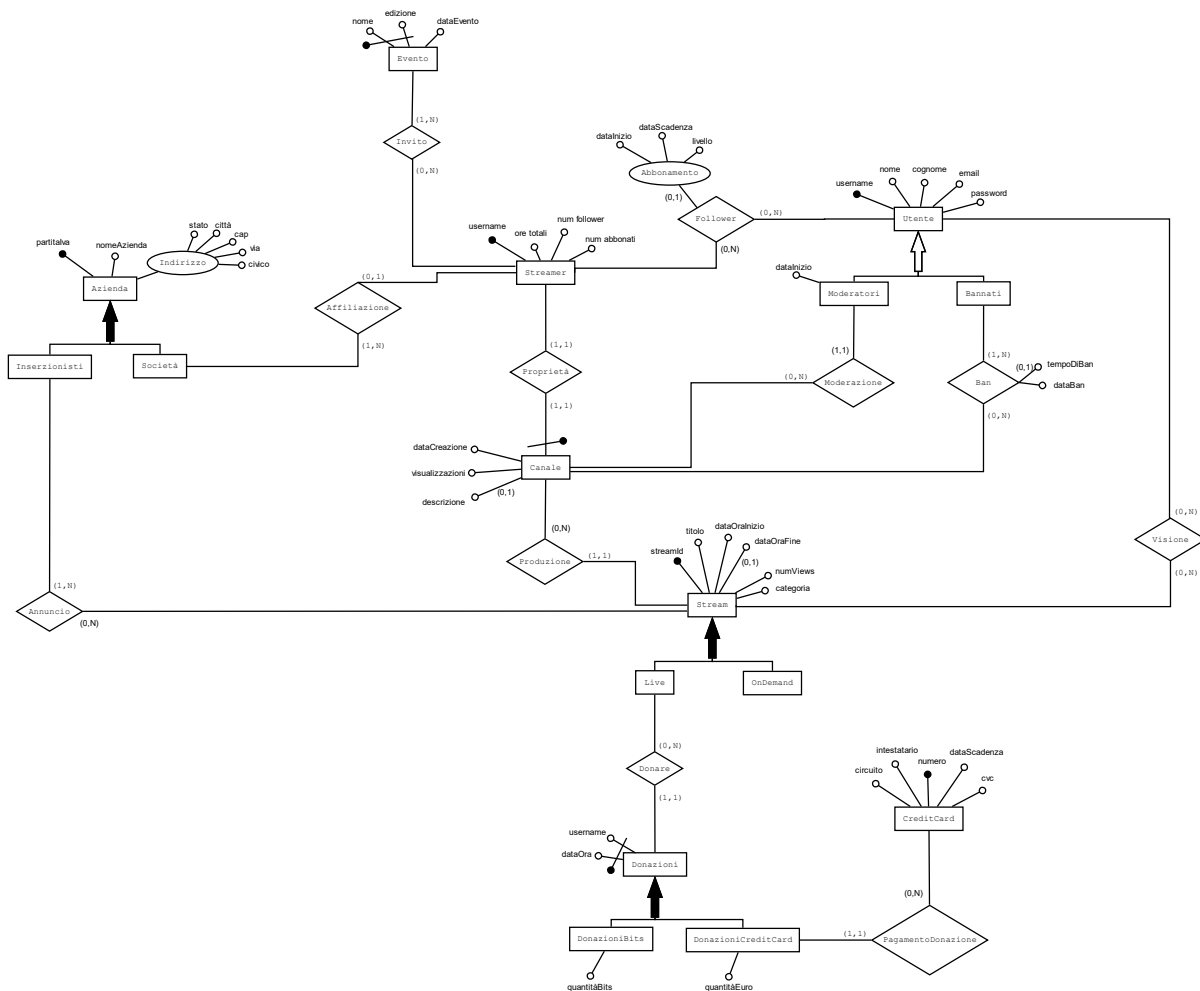
- Il numero di visualizzazioni di un canale deve essere uguale alla somma delle visualizzazioni di ogni stream archiviato in quel mese.
- Le ore totali di uno streamer sono uguali alla somma di (dataOraFine - dataOraInizio) di tutte le stream.
- Una Stream non può avere data/ora precedente alla data di creazione di un canale.
- Una donazione può avvenire solo se la sua data/ora sono comprese tra la data/ora d’inizio e fine della stream a cui è correlata.
- Un utente può essere moderatore di un canale solo se la sua data di inizio avviene dopo la data di creazione di quel canale.

Tabella delle relazioni:

RELAZIONE	ENTITÀ COINVOLTE	DESCRIZIONE	Attributi
Proprietà	Streamer(1,1) Canale(1,1)	Uno streamer possiede solo un canale Twitch, ed ogni canale può appartenere a un solo Streamer.	
Affiliazione	Streamer(0,1) Società(1,N)	Uno streamer può essere indipendente o far parte di una società. Una società gestisce almeno uno streamer.	
Follower	Streamer(0,N) Utente(0,N)	Uno streamer può avere un seguito o meno. Ogni utente può seguire uno o	Abbonamento

		più streamer.	
Invito	Streamer(0,N) Evento(1,N)	Uno streamer può essere invitato o meno a vari eventi. Ogni evento ha al minimo uno streamer invitato.	
Moderazione	Canale(0,N) Moderatori(1,1)	Un canale può avere o no moderatori, un moderatore è tale solo in un solo canale.	
Ban	Canale(0,N) Bannati(1,N)	Un canale può avere degli utenti bannati da esso. Un utente bannato lo può essere da più canali.	tempoDiBan, dataBan
Produzione	Canale(0,N) Stream(1,1)	Un canale rende disponibili o no delle stream. Una stream può essere ospitata in un solo canale.	
Visione	Stream(0,N) Utente(0,1)	Una stream può avere zero o più spettatori. Un utente può star visualizzando una stream o meno.	
Donare	Live(0,N) Donazione(1,1)	In una live possono essere effettuate o meno delle donazioni. Una donazione è fatta a una sola live.	
PagamentoDonazione	DonazioneConCreditCard(1,1) CreditCard(0,N)	Una donazione con soldi è fatta con una carta di credito. Un utente può fare molteplici donazioni con la stessa carta di credito.	
Annuncio	Stream(0,N) Inserzionisti(1,N)	In una stream possono apparire le pubblicità di almeno un inserzionista. Gli inserzionisti appaiono in almeno una stream.	

Schema concettuale



PROGETTAZIONE LOGICA

Analisi delle ridondanze:

L'attributo “**visualizzazioni**” dell'entità Canale è ridondante in quanto può essere calcolato facendo la somma del numero di visualizzazioni di ciascuna Stream.

Tavola dei Volumi		
Canale	E	10.000.000
Stream	E	400.000.000
Produzione	R	400.000.000

Per ogni Canale ci sono in media 40 Stream. I Volumi di Stream e Produzione quindi sono stati calcolati secondo le seguenti ipotesi: $10.000.000 \times 40 = 400.000.000$

Operazioni in cui la ridondanza ha un effetto:

- Controllo numero di visualizzazioni totali in un canale (144 volte al giorno)
- Inserimento stream (20000 volte al giorno)

Presenza ridondanza:

Concetto	Costrutto	Accessi	Tipo
Canale	Entità	1	L

Viene effettuato 1 accesso al giorno x 144 volte per leggere il numero di visualizzazioni di un canale (operazione 1).

Concetto	Costrutto	Accessi	Tipo
Stream	Entità	1	S
Produzione	Relazione	40	L
Canale	Entità	1	S

Vengono effettuati 1 accesso x 20000 al giorno per aggiungere una nuova Stream al database, 40 accessi x 20000 per poter calcolare il numero di visualizzazioni totale, 1 accesso x 20000 per poter aggiornare il numero di visualizzazioni totale sul canale (operazione 2).

Tenendo conto che gli accessi in scrittura contano il doppio rispetto a quelli in lettura, abbiamo il costo giornaliero di: $144 + (20000 \times 2) + (40 \times 20000) + (20000 \times 2) = 880144$

Assenza ridondanza:

Concetto	Costrutto	Accessi	Tipo
Produzione	Relazione	40	L

Vengono effettuati 40 accessi al giorno x 144 volte per calcolare il numero di visualizzazioni totale di un canale (operazione 1).

Concetto	Costrutto	Accessi	Tipo
Stream	Entità	1	S

Viene effettuato solo 1 accesso al giorno x 20000 volte dato che questa volta non dobbiamo aggiornare nessun attributo sul numero totale di visualizzazioni (operazione 2).

Tenendo conto che gli accessi in scrittura contano il doppio rispetto a quelli in lettura, abbiamo il costo giornaliero di: $(40 \times 144) + 20000 = 25760$

In conclusione l'analisi della ridondanza dimostra che conviene eliminare l'attributo "visualizzazioni".

Eliminazione delle generalizzazioni:

- Padre: Azienda (**generalizzazione totale**)
Figlie: Inserzionisti , Società

Risoluzione: L'entità padre viene accorpata nelle figlie. La scelta di questa soluzione è stata contribuita dal caso che vengono effettuate prevalentemente operazioni solo su istanze delle classi figlie, ignorando quindi l'effettiva utilità dell'entità padre.

I dati identificativi di un'azienda verranno duplicati in Inserzionisti e Società.

- Padre: Stream (**generalizzazione totale**)
Figlie: Live , OnDemand

Risoluzione: Le entità figlie vengono accorpate nel padre. La scelta di questa soluzione è stata contribuita dal fatto che facendo così l'accesso sarà contestuale agli attributi del padre e delle figlie e siccome le entità figlie non hanno attributi propri non ci sarà spreco di memoria per i valori nulli.

Viene aggiunto un attributo booleano "inLive" all'entità padre Stream, se inLive = true allora si riferisce a una stream live altrimenti si riferisce a una stream onDemand. La relazione "donare" ora coinvolge le entità Stream (0,N) e Donazioni (1,1).

- Padre: Donazioni (**generalizzazione totale**)
Figlie: DonazioniBits , DonazioniCreditCard

Risoluzione: Le entità figlie vengono accorpate nel padre. È stata fatta questa scelta perché le figlie rappresentano lo stesso concetto. L'unico contro sarà lo spreco di memoria per gli attributi con valore nullo, ma in confronto alla situazione generale questa soluzione è ideale.

Viene aggiunto un attributo booleano “conBits” all’entità padre Donazioni, se conBits = true allora si riferisce a una donazione che è avvenuta con la valuta della piattaforma, altrimenti si riferisce a una donazione avvenuta con la carta di credito. Verranno aggiunti in seguito gli attributi “quantitàBits” e “quantitàEuro” che potranno essere nulli. E la relazione “pagamentoDonazione” coinvolgerà le entità Donazioni (0,1) e CreditCard (0, N).

- Padre: Utente (**generalizzazione parziale**)
Figlie: Moderatori , Bannati

Risoluzione: Si utilizza una soluzione ibrida.

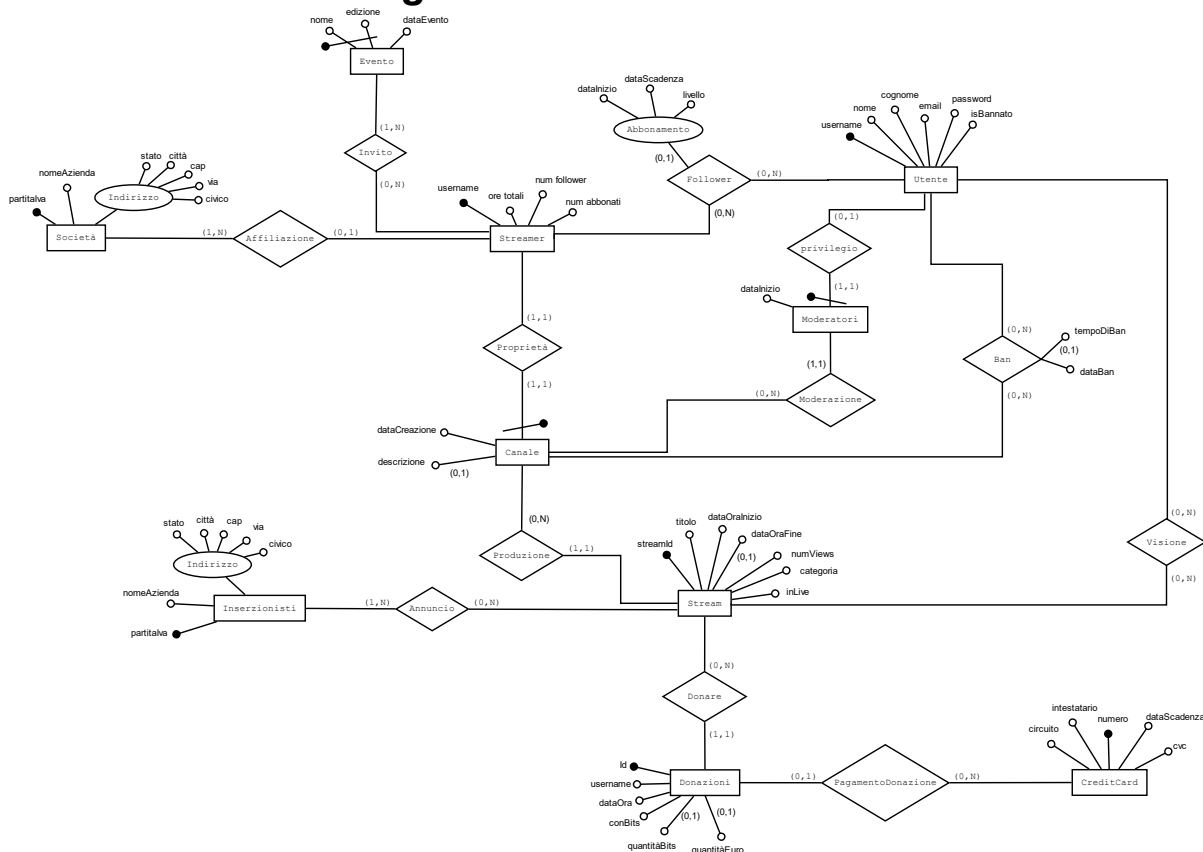
- L'entità figlia Bannati viene accorpata nel padre, in modo da avere accessi contestuali agli attributi del padre e delle figlie. Verrà aggiunto un ulteriore attributo booleano “isBannato”. La relazione “ban” coinvolgerà le entità Utente (0,N) e Canale (0,N).
- La generalizzazione con l'entità Moderatori viene sostituita con una relazione “privilegio” dato che gli accessi al padre non implicano accessi al figlio e viceversa. La relazione è chiave all'entità Moderatori.

Scelta degli identificatori primari:

La chiave di **Donazioni**, composta da due attributi (dataOra e username), diventa sconveniente in termini di memoria in quanto presenta tante occorrenze ed un elevato peso nella memoria (si parla di 12000 nuove istanze ogni giorno). Si è scelto per questo motivo quindi di usare come chiave di Donazioni un nuovo attributo Id.

Si era pensato poi di fare lo stesso pure con l’entità **Evento** ma siccome le occorrenze sono in numero minore rispetto alle altre entità/relazioni ed il suo peso in memoria non è poi così consistente si è deciso di mantenere la chiave attuale.

Diagramma E-R ristrutturato



Creazione delle tabelle: (A→B indica che B è chiave esterna di A)

Streamer(username, oreTotali, numFollower, numAbbonati, company→Societa.partitalva)

Canale(streamer→Streamer.username, dataCreazione, descrizione)

Stream(streamId, titolo, dataOrainizio, dataOraFine, numViews, categoria, inLive, canale→Canale.streamer)

Utente(username, nome, cognome, email, password, isBannato)

Follower(streamer→Streamer.username, utente→Utente.username, isAbbonato, dataInizio, dataScadenza, livello)

Moderatori(utente→Utente.username, dataInizio, canale→Canale.streamer)

Societa(partitalva, nomeAzienda, stato, citta, cap, via, civico)

Inserzionisti(partitalva, nomeAzienda, stato, citta, cap, via, civico)

Donazioni(Id, username→Utente.username, dataOra, conBits, quantitaBits, quantitaEuro,
stream→Stream.streamId, cc→CreditCard.numero)

CreditCard(numero, dataScadenza, cvc, intestatario, circuito)

Evento(nome, edizione, dataEvento)

ListaInvitati(nomeEvento→Evento.nome, edizioneEvento→Evento.edizione, invitato→Streamer.username)

ListaBannati(utente→Utente.username, canale→Canale.streamer, tempoDiBan, dataBan)

ListaAnnunci(inserzionista→Inserzionisti.partitalva, stream→Stream.streamId)

ListaVisioni(utente→Utente.username, stream→Stream.streamId)

QUERY E INDICI

Query:

1. Mostrare tutti i moderatori di un canale che ci sono stati fin dalla creazione di quest'ultima.

```
select dataCreazione, username
from Moderatori join Canale on Moderatori.Canale='watsonamelia_hololiveen'
where dataInizio=Canale.dataCreazione;
```

	datacreazione date	username character varying (25)
1	2022-04-01	Pufu22

2. Mostrare la top 10 degli streamer con più follower e che siano affiliati con una società con stabilimento in Giappone.

```
select username, numFollower
from Streamer
where company in(select partitalva from Società where Stato='Japan')
order by numFollower desc LIMIT 10;
```

	username [PK] character varying (25)	numfollower integer
1	gawrgura	3001201
2	Mumei	1231252
3	takanashikiara	600042
4	Bae	593041
5	Irys	500231
6	Fauna	412321
7	Kronii	312349
8	moriCalliope	200000
9	Ina	142415
10	Sana	100200

3. Stampare tutti i streamer che sono stati invitati negli eventi del 2020.

```
select username
from Streamer
where username in (select invitato from ListaInvitati where nomeEvento
                    in (select nome from Evento where dataEvento between
                        '2020-01-01' and '2020-12-31')
                    And edizioneEvento in (select edizione from Evento where
                                            dataEvento between '2020-01-01' and
                                            '2020-12-31'));
```

	username [PK] character varying (25)
1	veibae
2	shroud

4. Mostrare l'utente che ha fatto più donazioni a un certo streamer.

```
drop view if exists donatori;
Create view donatori as
select username, count(*) as numDon
from Donazione
where stream in (select streamId from Stream where
                 canale=(select streamer from Canale where streamer='Gaules'))
group by username
order by numdon desc;

select username, numDon
from donatori LIMIT 1;
```

	username character varying (25)	numdon bigint
1	AnBuck	1

5. Stampare gli inserzionisti presenti in quelle stream dove ci sono almeno 5 donazioni e hanno un numero di visualizzazioni maggiore di 1000.

```
drop view if exists streams;
create view streams as
select streamId
from Stream join Donazione on Stream.streamId=Donazione.stream
where Stream.numViews>1000
group by streamId
having Count(*)>5;

select nomeAzienda
from Inserzionisti
where partitaIva in (select inserzionista from ListaAnnunci as l
                     join streams on l.Stream=streams.streamid);
```

	nomeazienda character varying (50)
1	Colgate
2	Ritemed
3	Yahoo

6. Mostrare tutti gli eventi dopo il 2020 e di ciascun evento l'invitato con più abbonati.

```
drop view if exists eventi21 cascade;
Create view eventi21 as
select nome, edizione
from Evento
where dataEvento>'2020-12-31';

drop view if exists invitati21 cascade;
Create view invitati21 as
select e.nome,e.edizione,invitato
from eventi21 as e join ListaInvitati as li on e.nome=li.nomeEvento and
e.edizione=li.edizioneEvento;

drop view if exists mostabb;
Create view mostabb as
select nome,edizione,max(numAbbonati) as pog
from Streamer join invitati21 on invitati21.invitato=Username
group by nome,edizione;

select Nome,Edizione,Username
from mostabb join Streamer on pog=numabbonati;
```

	nome character varying (50)	edizione character varying (20)	username character varying (25)
1	TwitchCon	2021	watsonamelia_hololiveen
2	eSport	Fortnite	Gaules
3	MusicContest	BelgiumEdition	MontanaBlack88
4	MusicContest	ItalyEdition	MOONMOON
5	MusicContest	UkraineEdition	MOONMOON
6	eSport	WWE	Yassuo
7	eSport	Fifa	Castro_1021
8	eSport	BrawlStars	DrDisrespect

Indici:

Il controllo del numero di follower e di abbonati di un streamer sono dati di lettura ad alta frequenza giornaliera e sui quali vengono spesso fatti dei controlli, si richiede perciò una certa velocità nella ricerca di tale relazione, mentre le operazioni di scrittura (aggiornamento dei dati) sono molto meno frequenti. È bene quindi indicizzare la tabella Streamer sugli attributi “numFollower” e “numAbbonati”.

```
Create Index Idx_numfollower on Streamer(numfollower);
Create Index Idx_numabbonati on Streamer(numabbonati);
```

CODICE CPP:

Il codice cpp presenta il main e due funzioni eseguiQuery e printQuery, che servono rispettivamente all'esecuzione delle query e alla stampa a video delle istanze risultanti dalla query.

Prima di compilare il programma bisogna ridefinire le costanti usate per l'accesso al database con quelle della macchina corrente.

Tali costanti sono: NOMEDB, HOST, USER, PASS, PORT.

Per compilare il programma basta eseguire il comando:

```
g++ -L dependencies/lib Query.cpp -o Query -lpq
```

assicurandoci di essere nella cartella giusta.

e per eseguirlo .\Query.exe

Il programma mostrerà a schermo 6 query da cui selezionarne una digitando il numero corrispondente. Nel caso si selezionino la query 1 o 4 verrà visualizzati a schermo

degli streamer da cui l'utente dovrà selezionarne uno digitando l'username (nel caso 1 si consiglia di inserire **"watsonamelia_hololiveen"** e nel caso 4 di inserire **"Gaules"**).