

Relazione Progetto Basi di Dati

A.A. 2020-2021

# **Base di dati per la gestione di una piattaforma di Streaming Video**

Studentesse:

Castellucci Lucia (MATRICOLA 915056)

Lombardi Alessia (MATRICOLA 934294)



<b>Indice</b>	
<b>1. Introduzione</b>	
1.1. Descrizione del dominio applicativo	3
<b>2. Analisi dei requisiti</b>	
2.1 Requisiti in linguaggio naturale	4
2.2 Estrazione dei concetti principali	5
<b>3. Progettazione concettuale</b>	
3.1 Gestione degli abbonamenti	8
3.1.1 Gestione dei soggetti	9
3.2 Gestione degli utenti e delle visioni	10
3.3 Gestione delle produzioni	10
3.3.1 Gestione dei tipi di produzioni	11
3.3.2 Gestione delle serie TV	12
3.4 Schema finale	13
<b>4. Progettazione logica</b>	
4.1 Stima del volume dei dati	15
4.2 Descrizione delle operazioni principali e stima della loro frequenza	16
4.3 Schemi di navigazione e tabelle degli accessi	17
4.4 Analisi delle ridondanze	21
4.5 Raffinamento dello schema	23
4.6 Traduzione delle entità e delle associazioni in relazioni	25
4.7 Schema relazionale finale	27
4.8 Costruzione delle tabelle del DB in linguaggio SQL	29
4.9 Traduzione delle operazioni in query SQL	37
4.9.1. Inserimento di un nuovo titolare account.	37
4.9.2. Inserimento di un nuovo utente	37
4.9.3. Cancellazione di un titolare account esistente	37
4.9.4. Cancellazione di un utente esistente.	37
4.9.5. Numero di iscrizioni del mese corrente.	37
4.9.6 Ricerca per genere di una serie.	37
4.9.7. Visualizzare il numero di stagioni e il numero di episodi per una data serie.	38
4.9.8. Elenco degli utenti relativi ad un dato account.	38
4.9.9. Visualizzare la lista degli utenti.	38
4.9.10. Individuare la serie TV più lunga presente sulla piattaforma.	38
4.9.11. Calcolare l'incasso dell'anno corrente.	38
4.9.12. Individuare l'orario nel quale vengono viste più produzioni durante la giornata.	38
4.9.13. Migliori (più viste) serie TV nella piattaforma.	39
4.9.14. Elenco delle ultime visioni effettuate.	39
<b>5. Applicazione</b>	
5.1 Descrizione dell'architettura dell'applicazione realizzata	40

# 1. Introduzione

## 1.1 Descrizione di massima del dominio applicativo

Il progetto consiste nella realizzazione di un RDBMS per la gestione di una piattaforma di streaming video. Lo streaming è permesso agli account che, tramite un titolare di quest'ultimo, sottoscrivono un abbonamento. L'abbonamento ha un costo, che deve essere pagato mensilmente, nel medesimo giorno del mese. Inoltre, ogni account può essere usato da più utenti, anche contemporaneamente, fino ad un massimo di 5, ognuno dei quali fruisce di un account in comune (utilizza dunque le stesse chiavi di accesso alla piattaforma). Ogni utente può visionare una produzione, che può essere un film o un episodio di una serie. Gli episodi sono organizzati in stagioni e le stagioni organizzate in serie tv. Il database immagazzina ogni visione di produzioni da parte degli utenti. Ogni produzione è caratterizzata da una o più case produttrici, da uno o più generi, da uno o più lingue, e da un cast composto da uno o più registi, e da uno o più attori.

## 2. Analisi dei requisiti

### 2.1 Requisiti in linguaggio naturale

Si intende sviluppare l'interfaccia amministratore di una piattaforma di streaming video con le seguenti caratteristiche:

Un imprenditore decide di commissionare la realizzazione di una piattaforma di streaming video. Il suo obiettivo è quello di poter gestire una moltitudine di <b>film e serie tv</b> e di metterle a disposizione del pubblico, dietro pagamento di una somma mensile.	1
L'accesso alla piattaforma è permesso solo dopo la creazione di un <b>account</b> , ossia dopo aver generato delle credenziali di accesso al sistema.	2
Diversamente dallo streaming dei video presenti nella piattaforma, che siano film o serie tv, che sarà possibile solo dopo l'inserimento di <b>dati anagrafici e fiscali</b> , e la conseguente sottoscrizione di un <b>abbonamento</b> .	3
Il <b>pagamento</b> della somma dovuta per la visione è da corrispondere mensilmente nello stesso giorno del mese, con un importo fisso, che viene determinato al momento della sottoscrizione dell'abbonamento.	4
Una volta creato un account, anche senza aver sottoscritto alcun abbonamento, sarà possibile, da parte dei fruitori del servizio, la creazione di <b>utenti</b> diversi riferiti allo stesso account, di modo da poter consentire fino a 5 accessi simultanei dallo stesso account, e di tener traccia separatamente delle visioni di ogni soggetto accede alla piattaforma usando le medesime credenziali di accesso.	5
Ogni utente, dunque, una volta che si trova all'interno della piattaforma può decidere di <b>guardare uno dei video</b> che la piattaforma mette a disposizione, che può dunque essere un film o un episodio di una serie. Viene minuziosamente tenuto traccia del suddetto momento, salvando, oltre che data e ora di inizio, anche il video specifico del quale si sta prendendo visione e del minuto nel quale ci si è interrotti.	6
Gli <b>episodi</b> delle serie tv sono organizzati in <b>stagioni</b> e le stagioni sono organizzate in <b>serie tv</b> , per poter rendere immediata all'utente l'ordine che deve osservare per poter seguire cronologicamente una serie tv.	7
Sia dei <b>film</b> , che delle serie tv, viene immagazzinata qualche informazione rilevante, quale la <b>casa che ha prodotto il video</b> , il <b>genere</b> a cui esso afferisce, le <b>lingue</b> disponibili per la visione, e qualche <b>attore</b> che ha recitato, oltre che la <b>regia</b> .	8
Siccome la legge lo impone, il database fornisce all'utente un'indicazione sul tipo di pubblico adatto alla visione dei film e delle serie tv.	9
	10
	11
	12
	13
	14
	15
	16
	17
	18
	19
	20
	21
	22
	23
	24
	25
	26
	27
	28
	29
	30

*Tabella 1.1.1 - Specifiche di progetto in linguaggio naturale.*

## 2.2 Estrazione dei concetti principali

A seguito della lettura e comprensione dei requisiti richiesti dal cliente, si procede sviluppando un testo che ne riassume tutti i concetti e in particolare ne estragga quelli principali, risultando essere in questo modo meglio fruibile per la realizzazione della base di dati. Si tiene conto delle seguenti correzioni di ambiguità:

Riga	Termine	Definizione	Termine utilizzato nel DB	Sinonimi
2	Film e serie tv	Prodotti messi a disposizione dalla piattaforma.	Produzioni	
4	Account	Insieme delle credenziali di accesso alla piattaforma.	Account	
7	Dati anagrafici e fiscali	Insieme dei dati necessari per poter sottoscrivere un abbonamento.	Titolare account	
8	Abbonamento	Abilitazione per poter visionare le produzioni	Abbonamento	
9	Pagamento	Corrispettivo dovuto per far sì che venga validato l'abbonamento e quindi che si possano visionare le produzioni	Pagamento	
13	Utenti	Insieme dei soggetti che utilizzano la piattaforma dal medesimo account	Utente	Fruitori del servizio, clienti
18	Guardare uno dei video	Prendere visione di una delle produzioni disponibili sulla piattaforma	Visione	
22	Serie TV	Insieme di stagioni.	Serie TV	
22	Stagioni	Insieme di episodi.	Stagione	
22	Episodi	Tipo di produzione caratterizzata da un numero che ne indica l'ordine corretto di visione.	Episodio	
25	Film	Tipo di produzione.	Film	
26	Casa che ha prodotto il video	Casa produttrice della produzione.	Casa Produttrice	
26	Genere	Genere al quale appartiene la produzione.	Genere	
26	Lingua	Lingua nelle quali è distribuita la produzione.	Lingua	
27	Attore ... Regia	Soggetti che appartengono al cast di una produzione.	Cast	

**Tabella 1.1.2**–Tabella delle correzioni e glossario

Un imprenditore decide di commissionare la realizzazione di una piattaforma di streaming video. Il suo obiettivo è quello di poter gestire una moltitudine di <b>film e serie tv</b> . I fruitori del servizio possono accedere alla piattaforma creando un <b>account</b> , identificato da username e composto da una password. Per poter sottoscrivere un <b>abbonamento</b> al servizio, di cui si tiene traccia di un codice che lo identifica, di una data di inizio e di un prezzo, gli account devono essere associati ad uno ed un solo <b>titolare account</b> . Del titolare account è necessario inserire il CF che lo identifica, un indirizzo completo, e il numero della carta di credito. Esso è l'unico soggetto che può effettuare <b>pagamenti</b> riferiti all'abbonamento. Dei pagamenti si tiene traccia dell'importo e della data. Questi ultimi sono identificati univocamente dalla data e dall'abbonamento a cui si riferiscono.	1
La <b>visione</b> delle produzioni è possibile solo dagli utenti. Di questa si tiene traccia dell'ora di inizio, del minuto di arrivo, e dei minuti mancanti ed è identificata univocamente dalla data, dall'ora e dall'utente che la effettua. Gli <b>utenti</b> sono riferiti ad un account, che può avere al massimo 5 utenti connessi ad esso e sono caratterizzati da un nome utente, che li identifica in combinazione con l'account a cui fanno riferimento. Le <b>produzioni</b> possono essere di due diversi tipi, ossia di <b>film</b> o di <b>episodi</b> , e sono caratterizzate da un codice che le identifica, da un titolo, da un anno di uscita, da una durata e da un visto censura. Questi ultimi sono organizzati in <b>stagioni</b> , che a loro volta sono organizzate in <b>serie TV</b> . Le stagioni sono caratterizzate da un numero e identificate da esso e dalla serie a cui si riferiscono, che a sua volta è caratterizzata da un codice che la identifica, da un titolo e da una durata.	2
Viene immagazzinata inoltre, qualche informazione rilevante riguardo alle produzioni, quale la <b>casa produttrice</b> , identificata da un codice e caratterizzata dal nome e dall'anno di fondazione, il <b>genere</b> a cui esso afferisce, identificato da un codice e caratterizzato da una descrizione, le <b>lingue</b> disponibili per la visione, identificate da un codice e caratterizzate dalla nome della lingua stessa, e dal <b>cast</b> , che si compone di attori e registi, identificati da un codice e caratterizzati da nome, cognome. Degli attori è specificato anche il nome del personaggio. Ogni attore è inserito tante volte quante le produzioni in cui compare, a meno che non interpreti lo stesso personaggio in più produzioni.	3
	4
	5
	6
	7
	8
	9
	10
	11
	12
	13
	14
	15
	16
	17
	18
	19
	20
	21
	22
	23
	24
	25
	26
	27
	28

**Tabella 1.1.3-** Specifiche di progetto ristrutturate ed estrazione dei concetti principali.

Segue un elenco delle principali azioni richieste:

1. Inserimento di un nuovo titolare account.
2. Inserimento di un nuovo utente.
3. Cancellazione di un titolare account esistente.
4. Cancellazione di un utente esistente.
5. Numero di iscrizioni del mese corrente.

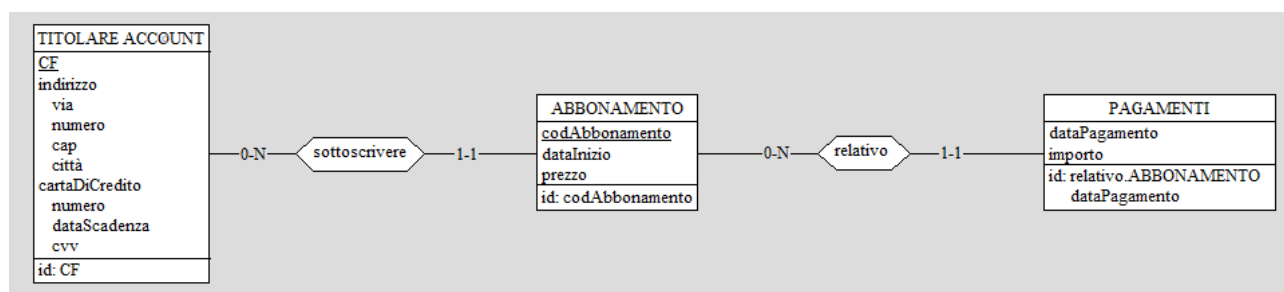
6. Ricerca per genere di una serie.
7. Visualizzare il numero di stagioni e il numero di episodi per una data serie.
8. Elenco degli utenti relativi ad un dato account.
9. Visualizzare la lista degli utenti.
10. Individuare la serie TV più lunga presente sulla piattaforma.
11. Calcolare l'incasso dell'anno corrente.
12. Individuare l'orario nel quale vengono viste più produzioni durante la giornata.
13. Migliori (più viste) serie TV nella piattaforma.
14. Elenco delle ultime visioni effettuate.

## 3. Progettazione concettuale

### 3.1 Gestione degli abbonamenti

#### Titolare account - Abbonamento

Dall'analisi del dominio è chiaro che l'entità titolare account è l'unica che può essere associata agli abbonamenti, siccome è l'unica che contiene dati fiscali e relativi alla carta di credito. L'entità titolare account è dunque identificata da un codice fiscale (si è deciso di ignorare il problema dell'omocodia), e caratterizzata da indirizzo (e i relativi attributi che lo compongono) e carta di credito (e i relativi attributi che lo compongono). Il titolare account è in associazione one-to-many con abbonamenti, dunque è possibile che un titolare account possa fare più abbonamenti. Ad esempio, un titolare potrebbe disdire l'abbonamento e decidere di rifarlo, oppure di avere più abbonamenti contemporaneamente per poter permettere a più di 5 utenti alla volta di utilizzare il servizio. Una volta che l'abbonamento viene disdetto, si è deciso di non mantenere uno storico, infatti abbonamento, identificato da un codice, è caratterizzato da una data di inizio, ma non da una data di fine, oltre che da un prezzo. Abbonamento ha partecipazione opzionale nell'associazione, mentre titolare account obbligatoria, infatti è possibile che un titolare disdica un abbonamento e non sia collegato ad alcuno di essi, mentre un abbonamento deve per forza avere un relativo titolare per esistere.



*Figura 1.1—Schema E/R con le principali entità per modellazione degli abbonamenti*

#### Abbonamento - Pagamenti

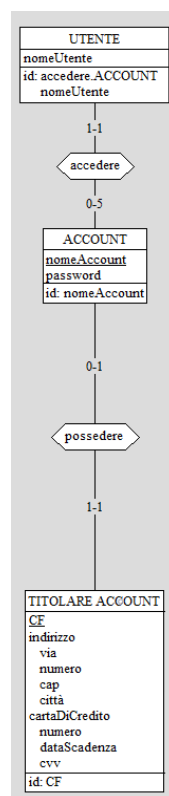
Dall'analisi del dominio si evince che abbonamento, essendo i pagamenti a cadenza mensile, sono in associazione one-to-many con questi ultimi. I pagamenti sono infatti caratterizzati da importo e identificati dalla data e dall'abbonamento a cui fanno riferimento. Pagamento ha partecipazione opzionale nell'associazione, mentre abbonamento obbligatoria, infatti è possibile che un abbonamento non sia stato pagato e che quindi ad esso non corrisponda alcun pagamento, mentre un pagamento deve per forza avere un relativo abbonamento per esistere.



### 3.1.1 Gestione dei soggetti della piattaforma

#### Titolare account - Account

Come esplicitato nell'analisi del dominio applicativo, un titolare account corrisponde ad uno ed un solo account, per cui l'associazione tra essi è one-to-one. Il primo è a partecipazione obbligatoria nell'associazione, mentre il secondo, identificato da un nome account e caratterizzato da una password, è a partecipazione opzionale, siccome il primo step per l'iscrizione alla piattaforma è quello di creare un account, mentre un titolare account può essergli associato in un secondo momento.



*Figura 1.2 – Schema E/R con le principali entità per modellazione dei soggetti interni al sistema*

#### Account - Utente

Come introdotto nell'analisi del dominio applicativo, ad un account possono corrispondere fino a 5 utenti, per cui l'associazione tra essi è one-to-many. Il primo è a partecipazione obbligatoria nell'associazione, mentre il secondo, identificato da un nome utente e dal nome del relativo account, è a partecipazione opzionale, siccome il primo step per l'iscrizione alla piattaforma è quello di creare un account, mentre gli utenti possono essergli associati in un secondo momento.

## 3.2 Gestione degli utenti e delle visioni

### Utente - Visione

Dall'analisi del dominio si evince che gli utenti possono effettuare più visioni, per cui le due entità sono in associazione one-to-many. Il primo è a partecipazione obbligatoria nell'associazione, mentre il secondo, identificato dall'utente che sta effettuando la visione, dalla data e dall'ora e caratterizzato da un minuto di arrivo, è a partecipazione opzionale, siccome è possibile che ci siano utenti che non hanno ancora visionato nulla, ma una visione per esistere deve riguardare una produzione.

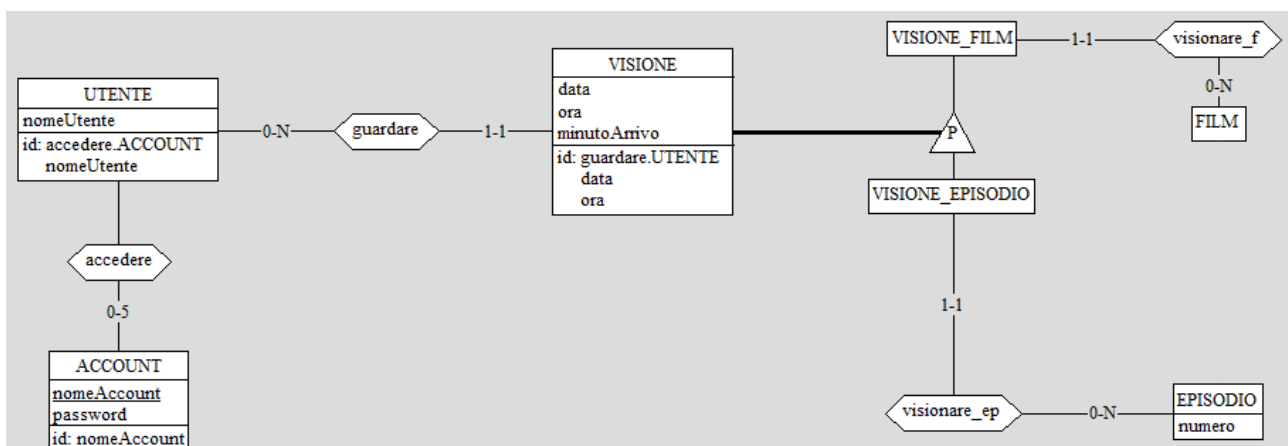


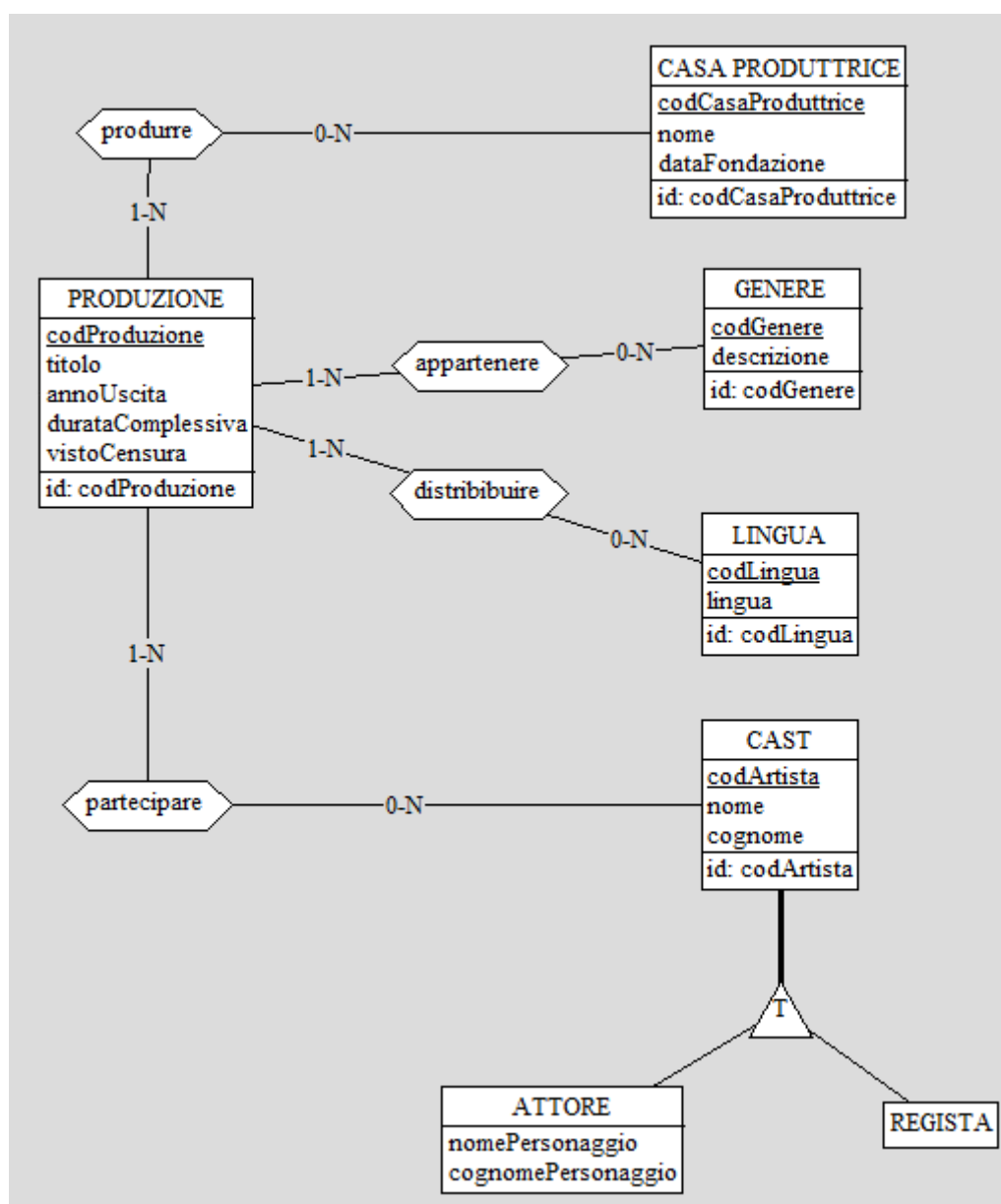
Figura 1.3 – Schema E/R con le principali entità per modellazione delle visioni

### Visione – Film/Episodio

Visione può essere di due tipi, o una visione di film, o una visione di episodio, ma di null'altro; si tratta dunque di una gerarchia totale e disgiunta. Visione è in associazione one-to-many con i due tipi di produzione, infatti è possibile che una produzione sia visionata più volte, mentre una visione riguarda una sola produzione. Visione prima è a partecipazione opzionale nell'associazione, mentre la produzione è a partecipazione obbligatoria, siccome una visione non esiste se non ci sono produzioni, mentre è vero il viceversa.

## 3.3 Gestione delle produzioni

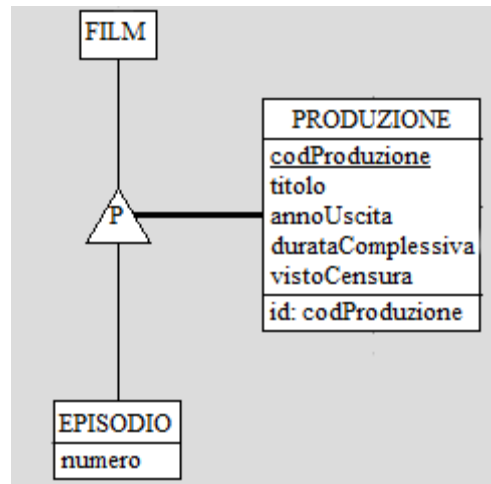
Come esplicitato nel dominio applicativo, delle produzioni sono state inserite diverse informazioni interessanti quali casa produttrice, lingua di distribuzione e genere a cui appartiene. Tutte queste entità sono in associazione many-to-many con produzione, siccome una produzione può essere prodotta da più case, distribuita in più lingue e appartenere a più generi e viceversa. Le varie entità sono a partecipazione obbligatoria mentre produzioni è a partecipazione opzionale.



*Figura 1.4–Schema E/R con le principali entità per modellazione delle produzioni*

### 3.3.1 Gestione dei tipi di produzione

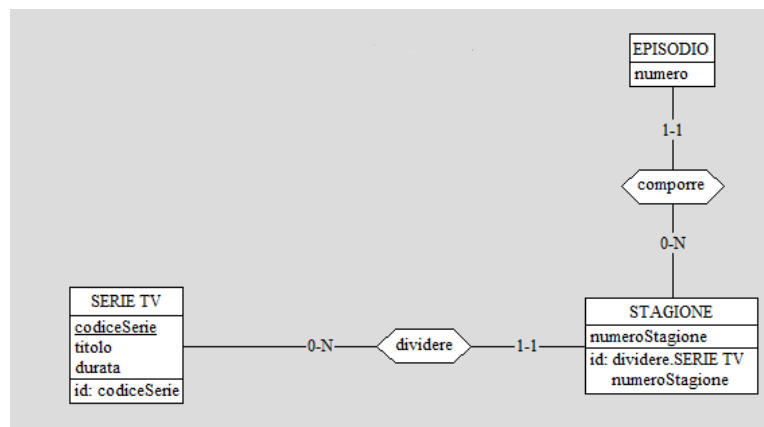
Produzione è identificata da un codice e caratterizzata da un titolo, da un anno di uscita, da una durata complessiva e da un visto censura. Può essere un episodio, o una serie, o null'altro. La gerarchia è dunque totale ed esclusiva. Episodio è caratterizzato ulteriormente da un numero.



**Figura 1.5** – Schema E/R con le principali entità per modellazione dei tipi di produzione

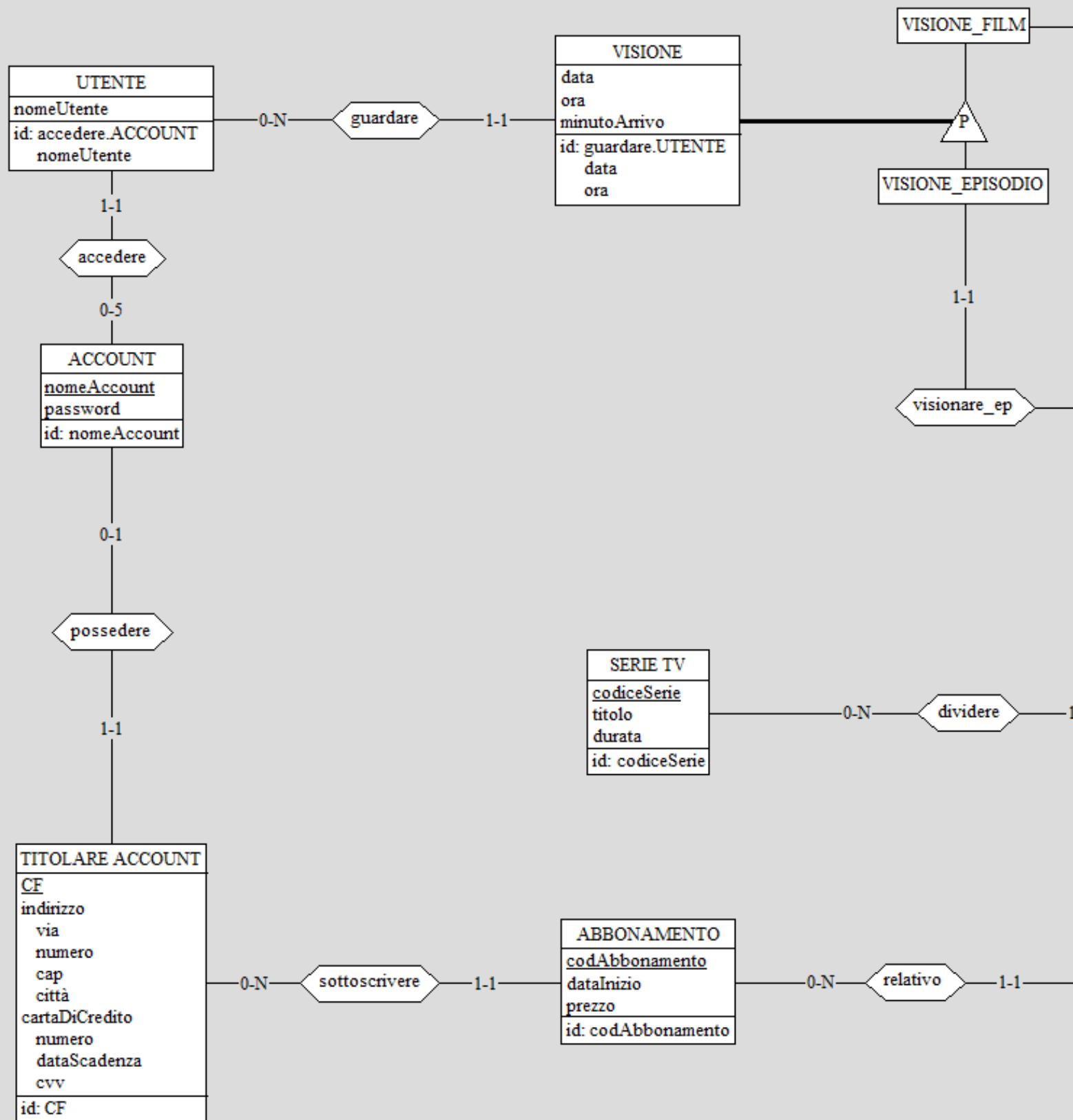
### 3.3.2 Gestione delle serie tv

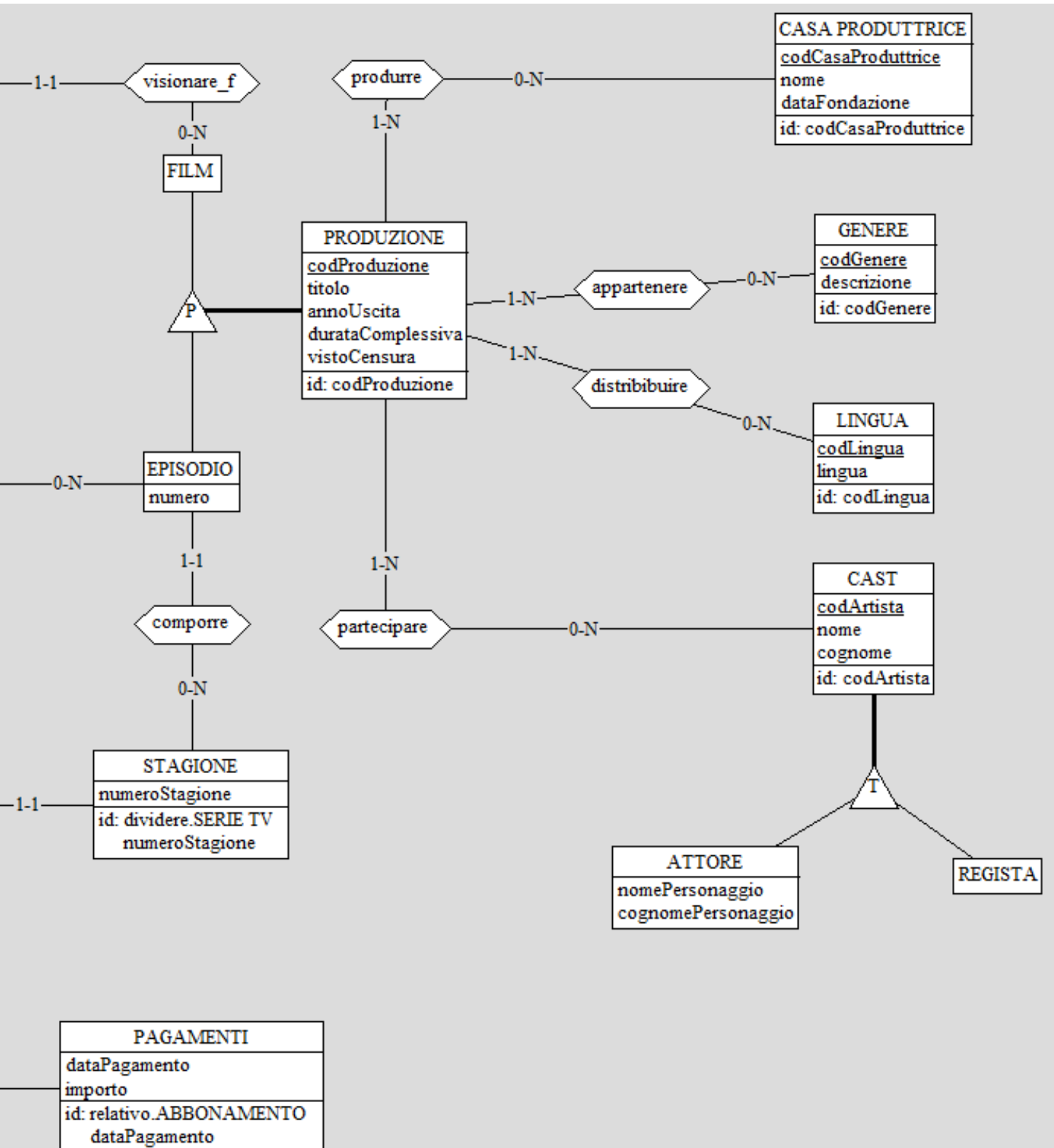
Come evidenziato dall'analisi del dominio, episodio è raggruppato in stagioni, per cui tra di essi vige un'associazione one-to-many. Le stagioni, identificate dal numero stagione e dalla serie TV a cui fanno riferimento e sono in associazione one-to-many con quest'ultima essendo essa un raggruppamento di stagioni.



**Figura 1.6** – Schema E/R con le principali entità per modellazione delle serie tv

### 3.2 Schema finale





## 4. Progettazione logica

### 4.1 Stima del volume dei dati

Concetto	Costrutto	Volume
Pagamento	E	1000
Relativo	R	1000
Abbonamento	E	100
Sottoscrivere	R	100
Titolare account	E	100
Possedere	R	100
Account	E	100
Accedere	R	300
Utente	E	300
Guardare	R	400000
Visione	E	400000
Visionare_f	R	35000
Film	E	150
Visionare_ep	R	365000
Episodio	E	1500
Comporre	R	1500
Stagione	E	100
Dividere	R	100
Serie	E	50
Casa produttrice	E	300
Genere	E	13
Lingua	E	10
Cast	E	6600
Produrre (riferito a film)	R	225
Produrre (riferito ad episodio)	R	2250
Appartenere (riferito a film)	R	375
Appartenere (riferito ad episodio)	R	4500
Distribuire (riferito a film)	R	600
Distribuire (riferito ad episodio)	R	6000
Partecipare (riferito a film)	R	600
Partecipare (riferito ad episodio)	R	6000

**Tabella 2.1.1-** Stima del volume dei dati.

## 4.2 Descrizione delle operazioni principali e stima della loro frequenza

Le operazioni da effettuare vengono ripetute, siccome già introdotte nella fase di analisi. Segue una tabella riportante la loro descrizione e relativa frequenza:

Codice	Descrizione	Frequenza
01	Inserimento di un nuovo titolare account	7/mese
02	Inserimento di un nuovo utente	10/mese
03	Cancellazione di un titolare account esistente	1/mese
04	Cancellazione di un utente esistente	1/mese
05	Numero di iscrizioni del mese corrente	1/mese
06	Ricerca per genere di una serie	2/anno
07	Visualizzare il numero di stagioni e il numero di episodi per una data serie	1/anno
08	Elenco degli utenti relativi ad un dato account	1/mese
09	Visualizzare la lista degli utenti	1/mese
10	Individuare la serie TV più lunga presente sulla piattaforma	1/mese
11	Calcolare l'incasso dell'anno corrente	6/anno
12	Individuare l'orario nel quale vengono viste più produzioni durante la giornata	1/settimana
13	Migliori (più viste) serie TV nella piattaforma	1/settimana
14	Elenco delle ultime visioni effettuate	1/giorno

**Tabella 2.1.2-** Descrizione delle operazioni principali e stima della loro frequenza.



## 4.3 Schemi di navigazione e tabelle degli accessi

Di seguito sono riportate le tabelle degli accessi delle operazioni sopra citate, e, dove non risulta banale, viene riportato anche lo schema di navigazione:

Segue un elenco delle principali azioni richieste:

### OPERAZIONE 01 – INSERIMENTO DI UN NUOVO TITOLARE ACCOUNT

Concetto	Costrutto	Accesso	Tipo
TITOLARE_ACCOUNT	E	1	S
ACCOUNT	E	1	S
POSSEDERE	R	1	S
TOT: 3S -> 7/mese			

### OPERAZIONE 02 – INSERIMENTO DI UN NUOVO UTENTE

Concetto	Costrutto	Accesso	Tipo
UTENTE	E	1	S
ACCEDERE	E	1	S
TOT: 2S -> 10/mese			

### OPERAZIONE 03 – CANCELLAZIONE DI UN TITOLARE ACCOUNT ESISTENTE

In media ci sono 3 utenti per ogni account, per cui:

Concetto	Costrutto	Accesso	Tipo
ACCOUNT	E	1	S
ACCOUNT	E	1	L
POSSEDERE	R	1	S
POSSEDERE	R	1	L
TITOLARE_ACCOUNT	E	1	S
TITOLARE_ACCOUNT	E	1	L
ABBONAMENTO	E	1	S
ABBONAMENTO	E	1	L
SOTTOSCRIVERE	R	1	S
SOTTOSCRIVERE	R	1	L
ACCEDERE	R	3	S
ACCEDERE	R	3	L
UTENTE	E	3	S
UTENTE	E	3	L

TOT: 7L + 7S -> 1/mese

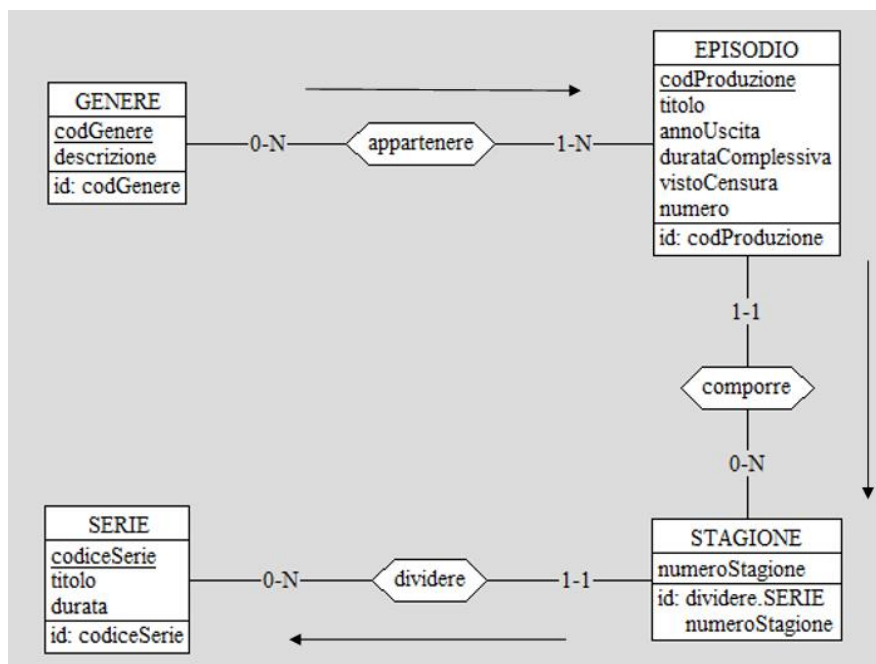
#### OPERAZIONE 04 – CANCELLAZIONE DI UN UTENTE ESISTENTE

Concetto	Costrutto	Accesso	Tipo
UTENTE	E	I	L
UTENTE	E	I	S
ACCEDERE	R	I	L
ACCEDERE	R	I	S
TOT: 2L + 2S -> 1/mese			

#### OPERAZIONE 05 – NUMERO DI ISCRITTI DEL MESE CORRENTE

Concetto	Costrutto	Accesso	Tipo
ABBONAMENTO	E	100	L
TOT: 100L -> 1/mese			

#### OPERAZIONE 06 – RICERCA PER GENERE DI UNA SERIE



In media ci sono 346 episodi per genere, e conseguentemente, siccome in media ci sono 15 episodi per ogni stagione, in questo caso abbiamo 23 stagioni. Inoltre, siccome in media ci sono 2 stagioni per serie, in questo caso abbiamo circa 11 serie. Dunque, calcolo il costo in questa maniera:

Concetto	Costrutto	Accesso	Tipo
GENERE	E	I	L

appartenere_ep	R	346	L
EPISODIO	E	346	L
COMPORRE	R	346	L
STAGIONE	E	23	L
DIVIDERE	R	23	L
SERIE	E	11	L
TOT: 1084L -> 2/anno			

#### OPERAZIONE 07 – VISUALIZZARE IL NUMERO DI STAGIONI E IL NUMERO DI EPISODI PER UNA DATA SERIE

In media ci sono 2 stagioni per serie e 15 episodi per ogni stagione per un totale di 30 episodi per serie. Dunque, calcolo il costo in questa maniera:

Concetto	Costrutto	Accesso	Tipo
SERIE	E	1	L
DIVIDERE	R	2	L
STAGIONE	E	2	L
COMPORRE	R	30	L
EPISODIO	E	30	L
TOT: 65L -> 1/anno			

#### OPERAZIONE 08 – ELENCO DEGLI UTENTI RELATIVI AD UN DATO ACCOUNT

In media ci sono 3 utenti in associazione con ogni account. Dunque, calcolo il costo in questa maniera:

Concetto	Costrutto	Accesso	Tipo
UTENTE	E	3	L
ACCEDERE	R	3	L
ACCOUNT	E	1	L
TOT: 7L -> 1/mese			

#### OPERAZIONE 09 – VISUALIZZARE LA LISTA DEGLI UTENTI

Concetto	Costrutto	Accesso	Tipo
UTENTE	E	300	L
TOT: 300L -> 1/mese			

#### OPERAZIONE 10 – INDIVIDUARE LA SERIE TV PIU' LUNGA PRESENTE SULLA PIATTAFORMA

Concetto	Costrutto	Accesso	Tipo
SERIE	E	50	L

TOT: 50L -> 1/mese

## OPERAZIONE 11 – CALCOLARE L'INCASSO DELL'ANNO CORRENTE

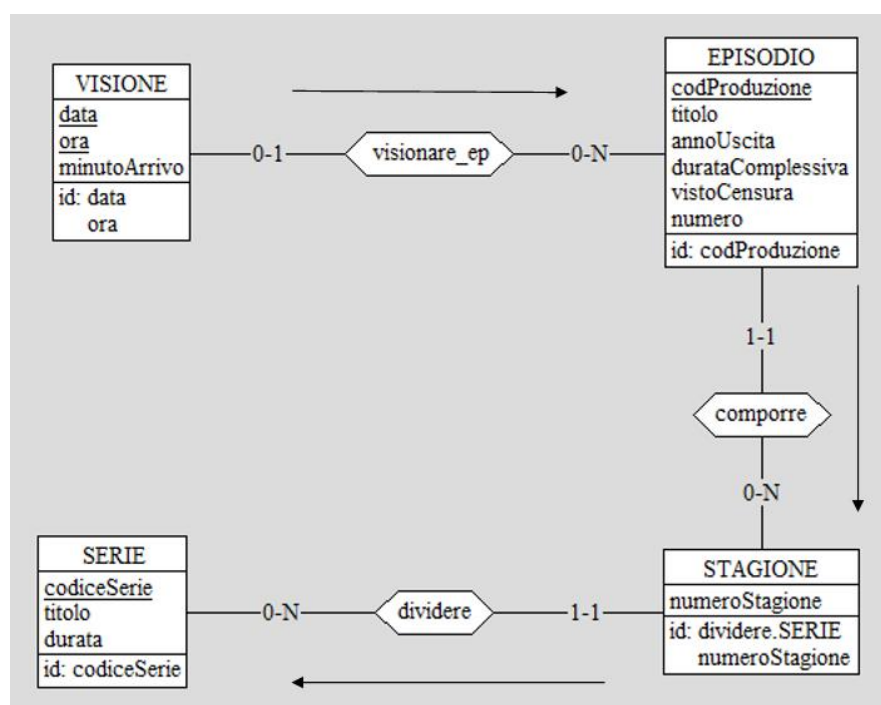
Siccome i primi pagamenti risalgono a gennaio 2020, si è fatta una media di pagamenti per mese, per cui dividendo i pagamenti totali per 18 mesi (ad arrivare ad oggi), e si sono moltiplicati per 12 (anno solare).  
Dunque, calcolo il costo in questa maniera:

Concetto	Costrutto	Accesso	Tipo
PAGAMENTO	E	665	L
TOT: 665L -> 6/anno			

## OPERAZIONE 12 – INDIVIDUARE L'ORARIO NEL QUALE VENGONO VISTE PIU' PRODUZIONI DURANTE LA GIORNATA

Concetto	Costrutto	Accesso	Tipo
VISIONE	E	400000	L
TOT: 400000L -> 1/SETTIMANA			

## OPERAZIONE 13 – MIGLIORI (PIU' VISTE) SERIE TV NELLA PIATTAFORMA



Concetto	Costrutto	Accesso	Tipo
VISIONE	E	365000	L
VISIONARE_EP	R	365000	L
EPISODIO	E	1500	L
COMPORRE	R	1500	L

STAGIONE	E	100	L
DIVIDERE	R	100	L
SERIE	E	50	L
TOT: 733250L -> 1/settimana			

#### OPERAZIONE 14 – ELENCO DELLE ULTIME VISIONI EFFETTUATE

Siccome le prime visioni risalgono a gennaio 2020, si è fatta una media di visioni per giorno, per cui dividendo le visioni totali per 540 giorni (ad arrivare ad oggi). Dunque, calcolo il costo in questa maniera:

Concetto	Costrutto	Accesso	Tipo
VISIONE	E	740	L
TOT: 740L -> 1/GIORNO			

## 4.4 Analisi delle ridondanze

È stata inserito un campo ridondante "durata" nella tabella serie, che lo si sarebbe potuto calcolare sommando le durate dei singoli episodi. L'unica operazione che interessa questo campo è la numero 10, che individua la serie più lunga sulla piattaforma. Di seguito la valutazione nel costo nel caso in cui si voglia mantenere la ridondanza:

OPERAZIONE 10 - <b>CON</b> CAMPO durata IN SERIE RIDONDANTE			
Concetto	Costrutto	Accesso	Tipo
SERIE	E	50	L
TOT: 50L -> 1/mese			

Di seguito la valutazione nel costo nel caso in cui si voglia eliminare la ridondanza con la relativa tabella degli accessi:

OPERAZIONE 10 - <b>SENZA</b> CAMPO durata IN SERIE RIDONDANTE			
Concetto	Costrutto	Accesso	Tipo
EPISODIO	E	1500	L
COMPORRE	R	1500	L
STAGIONE	E	100	L
DIVIDERE	R	100	L
SERIE	E	50	L
TOT: 3250L -> 1/mese			

È stata inserito un campo ridondante "importo" nella tabella pagamento, che lo si sarebbe potuto calcolare sommando i prezzi degli abbonamenti pagati. L'unica operazione che interessa questo campo è la numero 11, che individua l'incasso totale dell'anno corrente. Di seguito la valutazione nel costo nel caso in cui si voglia mantenere la ridondanza:

OPERAZIONE 11 - <b>CON</b> CAMPO importo IN PAGAMENTO RIDONDANTE			
Concetto	Costrutto	Accesso	Tipo
PAGAMENTO	E	665	L
TOT: 665L -> 6/anno			

Di seguito la valutazione nel costo nel caso in cui si voglia eliminare la ridondanza:

OPERAZIONE 11 - <b>SENZA</b> CAMPO importo IN PAGAMENTO RIDONDANTE			
Concetto	Costrutto	Accesso	Tipo
PAGAMENTO	E	665	L
RELATIVO	R	665	L
ABBONAMENTO	E	665	L
TOT: 1995L -> 6/anno			

Si può vedere come in entrambi i casi la ridondanza garantisca un notevole vantaggio in termini di accessi risparmiati, per questo verrà mantenuta.

## 4.5 Raffinamento dello schema

### Eliminazione delle gerarchie

Per l'eliminazione della gerarchia **produzioni** si è scelto di adottare l'approccio del collasso verso il basso, replicando così gli attributi in film ed episodio. Si è adottata questa strategia siccome, non avendo alcun vincolo in quanto la copertura della gerarchia è totale ed esclusiva, si è trovato più semplice far interagire gli episodi con le stagioni e quindi con le serie, lasciandoli distinti dai film.

Per l'eliminazione della gerarchia **cast** si è scelto di adottare l'approccio del collasso verso l'alto. Per poterlo fare, considerando che la copertura della gerarchia è sovrapposta, è stato inserito un selettore per ogni sotto entità di cast, per cui uno per attore e uno per regista. Si è evitato di aggiungere un terzo selettore nel caso in cui fossero tutti falsi siccome non è possibile che si verifichi il caso in quanto la gerarchia ha copertura totale. Si è adottata questa strategia, rispetto al collasso verso il basso, siccome in quel caso si sarebbe introdotta ridondanza, in quanto un attore che facesse anche il regista sarebbe presente all'interno del database 2 volte.

Per l'eliminazione della gerarchia **visione** si è scelto di adottare l'approccio del collasso verso l'alto. Si è adottata questa strategia siccome, non avendo alcun vincolo in quanto la copertura della gerarchia è totale ed esclusiva, in quanto si ha la necessità che nella stessa data e ora la visione di una produzione potesse essere permessa ad un solo utente.

### Eliminazione degli attributi composti

Nello schema è presente l'attributo composito "Indirizzo" nell'entità titolare account che è stato diviso nei suoi sottocomponenti, e l'attributo composito "Carta di credito" del quale è stato conservato solo il numero della carta.

### Scelta delle chiavi primarie

Nello schema sono già evidenziate senza ambiguità tutte le chiavi primarie per tutte le entità.

### Eliminazione degli identificatori esterni

Nello schema E/R sono eliminate le seguenti relazioni:

- Relativo, importando il codAbbonamento di Abbonamento in Pagamento.
- Sottoscrivere, importando il CF di Titolare account in Abbonamento.
- Possedere, importando il nomeAccount di Account in Titolare account.
- Accedere, importando il nomeAccount di Account in Utente.
- Guardare, importando il nomeUtente e il nomeAccount di Utente in Visione.
- Visionare f: importando il codFilm di Film in Visione (come attribuito facoltativo).

- Visionare\_ep: importando il codEpisodio di Episodio in Visione (come attributo facoltativo).
- Dividere: importando il codiceSerie di Serie in Stagione.
- Comporre: importando il numeroStagione e il codiceSerie di Stagione in Episodio.
- Produrre (in associazione con Film):reificata importando codFilm da Film e codCasaProduttrice da Casa produttrice.
- Produrre (in associazione con Episodio):reificata importando codEpisodio da Episodio e codCasaProduttrice da Casa produttrice.
- Appartenere (in associazione con Film):reificata importando codFilm da Film e codGenere da Genere,
- Appartenere (in associazione con Episodio):reificata importando codEpisodio da Episodio e codGenere da Genere.
- Distribuire (in associazione con Film):reificata importando codFilm da Film e codLingua da Lingua.
- Distribuire (in associazione con Episodio):reificata importando codEpisodio da Episodio e codLingua da Lingua.
- Partecipare (in associazione con Film):reificata importando codFilm da Film e codArtista da Cast.
- Partecipare (in associazione con Episodio):reificata importando codEpisodio da Episodio e codArtista da Cast.



## 4.6 Traduzione delle entità e delle associazioni in relazioni

**PAGAMENTO** (codAbbonamento, dataPagamento, importo)

FK: codAbbonamento REFERENCES ABBONAMENTO

**ABBONAMENTO** (codAbbonamento, dataInizio, prezzo, CF)

FK: CF REFERENCES TITOLARE\_ACCOUNT

**TITOLARE\_ACCOUNT**(CF, nomeAccount, indirizzo, città, numCartaDiCredito)

FK: nomeAccount REFERENCES ACCOUNT

**ACCOUNT**(nomeAccount, password)

**UTENTE**(nomeAccount, nomeUtente)

FK: nomeAccount REFERENCES ACCOUNT

**VISIONE**(nomeAccount, nomeUtente, data, ora, minutoArrivo, codFilm, codEpisodio)

FK: nomeAccount REFERENCES UTENTE

FK: nomeUtente REFERENCES UTENTE

**FILM**(codFilm, titolo, annoUscita, durataComplessiva, vistoCensura)

**EPISODIO**(codEpisodio, titolo, annoUscita, durataComplessiva, vistoCensura, numero, codiceSerie, numerostagione)

FK: codiceSerie REFERENCES STAGIONE

FK: numeroStagione REFERENCES STAGIONE

**STAGIONE**(numeroStagione, codiceSerie)

FK: codiceSerie REFERENCES SERIE

**SERIE**(codiceSerie, titolo, durata)

**CASA\_PRODUTTRICE**(codCasaProduttrice, nome, annoFondazione)

**GENERE**(codGenere, descrizione)

**LINGUA**(codLingua, lingua)

**CAST**(codArtista, nome, cognome, attore, regista, nomePersonaggio)

**produrre\_film**(codCasaProduttrice, codFilm)

FK: codCasaProduttrice REFERENCES CASA\_PRODUTTRICE

FK: codFilm REFERENCES FILM

**produrre\_ep**(codCasaProduttrice, codEpisodio)

FK: codCasaProduttrice REFERENCES CASA\_PRODUTTRICE

FK: codEpisodio REFERENCES EPISODIO

**appartenere\_film**(codGenere, codFilm)

FK: codGenere REFERENCES GENERE

FK: codFilm REFERENCES FILM

**appartenere\_ep**(codGenere, codEpisodio)

FK: codGenere REFERENCES GENERE

FK: codEpisodio REFERENCES EPISODIO

**distribuire\_film**(codLingua, codFilm)

FK: codLingua REFERENCES LINGUA

FK: codFilm REFERENCES FILM

**distribuire\_ep**(codLingua, codEpisodio)

FK: codLingua REFERENCES LINGUA

FK: codEpisodio REFERENCES EPISODIO

**partecipare\_film**(codArtista, codFilm)

FK: codArtista REFERENCES CAST

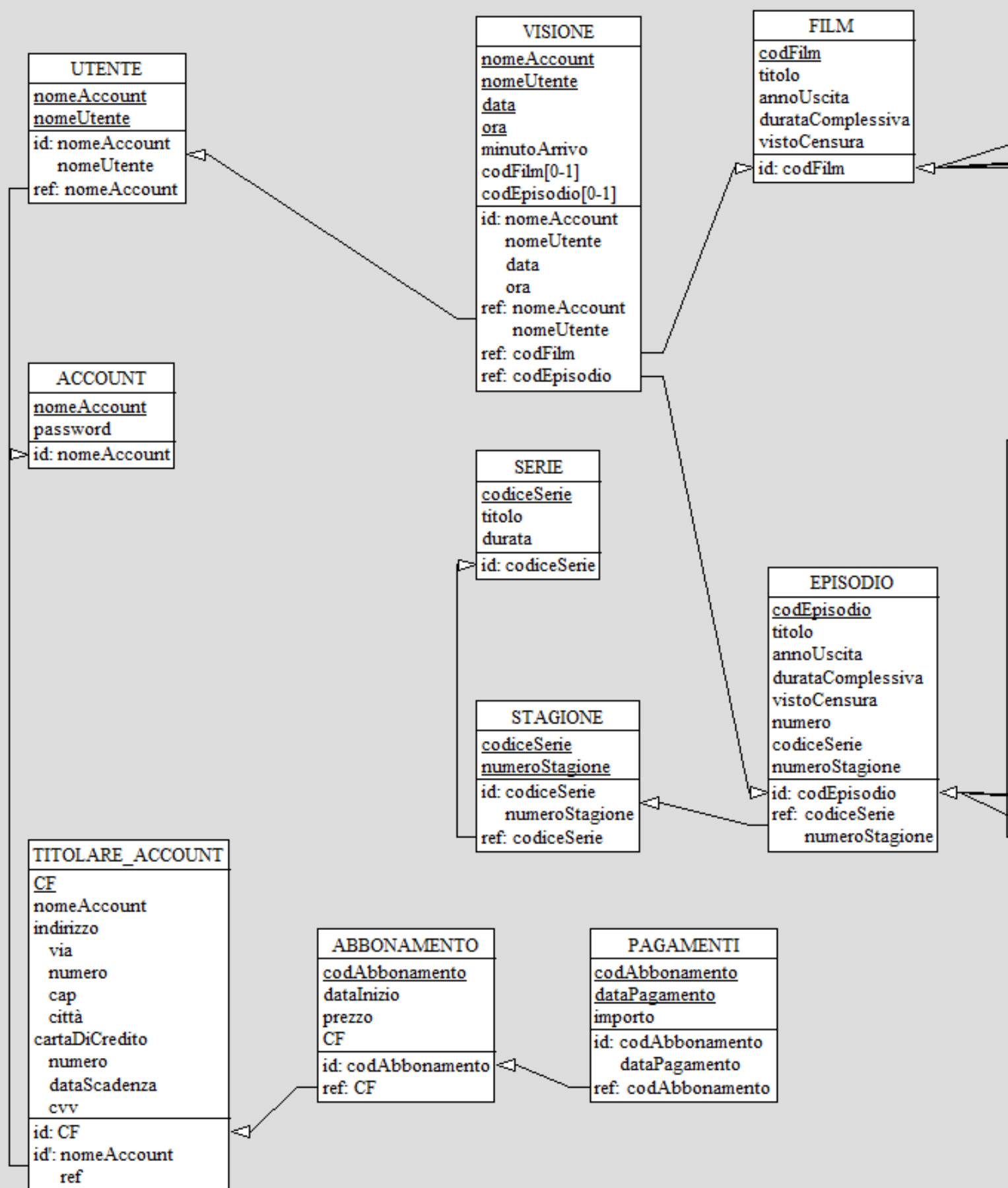
FK: codFilm REFERENCES FILM

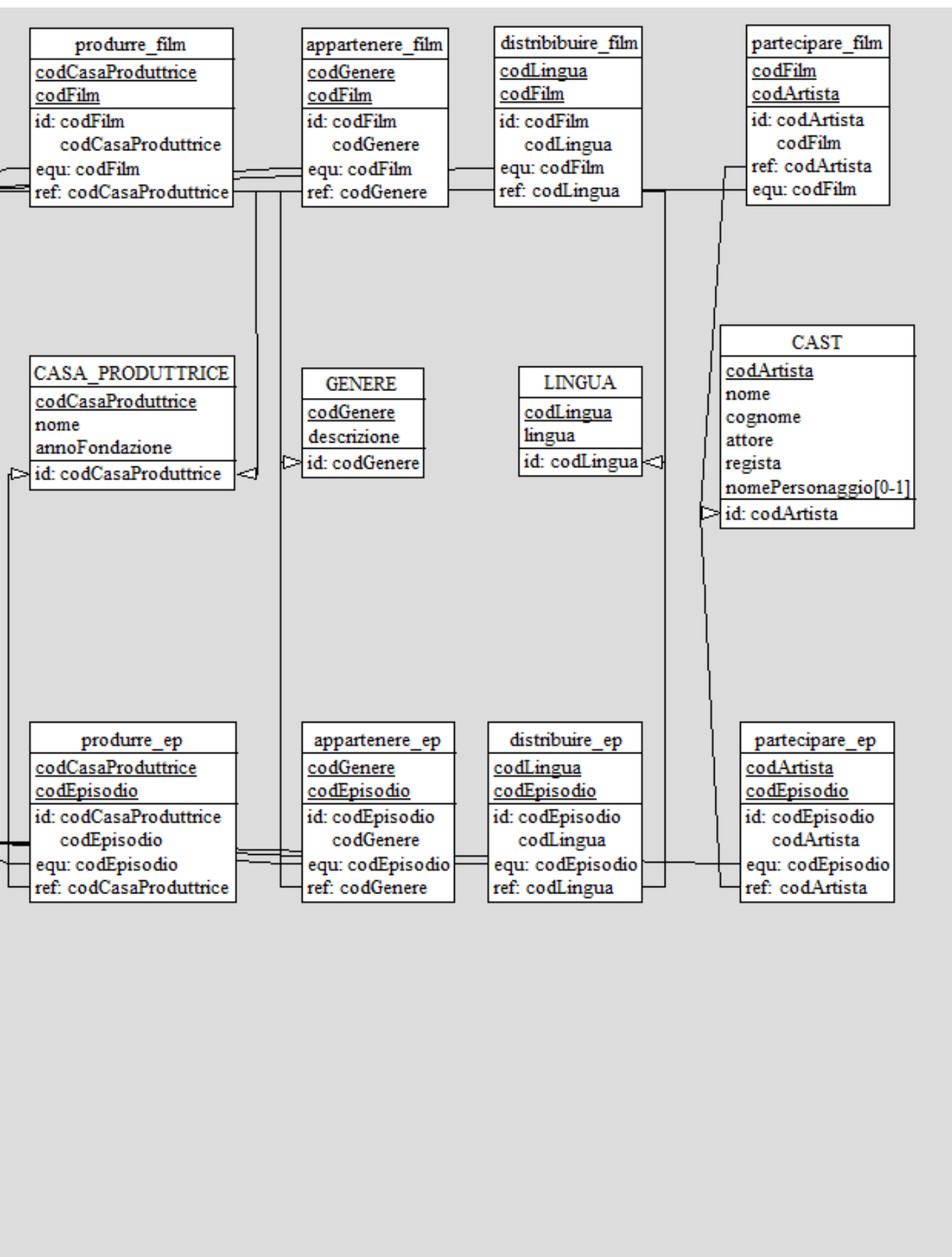
**partecipare\_ep**(codArtista, codEpisodio)

FK: codArtista REFERENCES CAST

FK: codEpisodio REFERENCES EPISODIO

## 4.7 Schema relazionale finale





## 4.8 Costruzione delle tabelle del DB in linguaggio SQL

### **- ABBONAMENTO**

```
create table ABBONAMENTO (  
    codAbbonamento varchar(255) not null,  
    dataInizio date not null,  
    prezzo float not null,  
    CF varchar(255) not null,  
    constraint IDABBONAMENTO primary key (codAbbonamento));
```

```
alter table ABBONAMENTO add constraint FKsottoscrivere  
    foreign key (CF)  
    references TITOLARE_ACCOUNT;
```

### **- ACCOUNT**

```
create table ACCOUNT (  
    nomeAccount varchar(255) not null,  
    password varchar(255) not null,  
    constraint IDACCOUNT primary key (nomeAccount));
```

### **- appartenere\_ep**

```
create table appartenere_ep (  
    codGenere varchar(255) not null,  
    codEpisodio varchar(255) not null,  
    constraint IDappartenere_ep primary key (codEpisodio, codGenere));
```

```
alter table appartenere_ep add constraint FKapp_EPI  
    foreign key (codEpisodio)  
    references EPISODIO;
```

```
alter table appartenere_ep add constraint FKapp_GEN  
    foreign key (codGenere)  
    references GENERE;
```

### **- appartenere\_film**

```
create table appartenere_film (  
  codGenere varchar(255) not null,  
  codFilm varchar(255) not null,  
  constraint IDappartenere_film primary key (codFilm, codGenere));
```

```
alter table appartenere_film add constraint FKapp_FIL  
  foreign key (codFilm)  
  references FILM;
```

```
alter table appartenere_film add constraint FKapp_GEN  
  foreign key (codGenere)  
  references GENERE;
```

#### - **CASA\_PRODUTTRICE**

```
create table CASA_PRODUTTRICE (  
  codCasaProduttrice varchar(255) not null,  
  nome varchar(255) not null,  
  annoFondazione int not null,  
  constraint IDCASA_PRODUTTRICE primary key (codCasaProduttrice));
```

#### - **CAST**

```
create table CAST (  
  codArtista varchar(255) not null,  
  nome varchar(255) not null,  
  cognome varchar(255) not null,  
  attore char not null,  
  regista char not null,  
  nomePersonaggio varchar(255),  
  constraint IDCAST primary key (codArtista));
```

#### - **distribuire\_film**

```
create table distribuire_film (  
  codLingua varchar(255) not null,  
  codFilm varchar(255) not null,  
  constraint IDdistribuire_film primary key (codFilm, codLingua));
```

```
alter table distribuire_film add constraint FKdis_FIL
foreign key (codFilm)
references FILM;
```

```
alter table distribuire_film add constraint FKdis_LIN
foreign key (codLingua)
references LINGUA;
```

#### - **distribuire\_ep**

```
create table distribuire_ep (
codLingua varchar(255) not null,
codEpisodio varchar(255) not null,
constraint IDdistribuire_ep primary key (codEpisodio, codLingua));
```

```
alter table distribuire_ep add constraint FKdis_EPI
foreign key (codEpisodio)
references EPISODIO;
```

```
alter table distribuire_ep add constraint FKdis_LIN
foreign key (codLingua)
references LINGUA;
```

#### - **EPISODIO**

```
create table EPISODIO (
codEpisodio varchar(255) not null,
titolo varchar(255) not null,
durataComplessiva time not null,
vistoCensura varchar(255) not null,
numero int not null,
annoUscita int not null,
codiceSerie varchar(255) not null,
numeroStagione int not null,
constraint IDEPISODIO_ID primary key (codEpisodio));
```

```
alter table EPISODIO add constraint FKR
foreign key (codiceSerie, numeroStagione)
```

references STAGIONE;

### - **FILM**

```
create table FILM (  
    codFilm varchar(255) not null,  
    titolo varchar(255) not null,  
    durataComplessiva time not null,  
    vistoCensura varchar(255) not null,  
    annoUscita int not null,  
    constraint IDFILM_ID primary key (codFilm));
```

### - **GENERE**

```
create table GENERE (  
    codGenere varchar(255) not null,  
    descrizione varchar(255) not null,  
    constraint IDGENERE primary key (codGenere));
```

### - **LINGUA**

```
create table LINGUA (  
    codLingua varchar(255) not null,  
    lingua varchar(255) not null,  
    constraint IDLINGUA primary key (codLingua));
```

### - **PAGAMENTO**

```
create table PAGAMENTO (  
    codAbbonamento varchar(255) not null,  
    dataPagamento date not null,  
    importo float not null,  
    constraint IDPAGAMENTO primary key (codAbbonamento, dataPagamento));
```

```
alter table PAGAMENTO add constraint FKrelativo  
    foreign key (codAbbonamento)  
references ABBONAMENTO;
```

### - **partecipare\_ep**



```
create table partecipare_ep (  
  codArtista varchar(255) not null,  
  codEpisodio varchar(255) not null,  
  constraint IDpartecipare_ep primary key (codEpisodio, codArtista));
```

```
alter table partecipare_ep add constraint FKpar_EPI  
  foreign key (codEpisodio)  
  references EPISODIO;
```

```
alter table partecipare_ep add constraint FKpar_CAS  
  foreign key (codArtista)  
  references CAST;
```

#### **- partecipare film**

```
create table partecipare_film (  
  codFilm varchar(255) not null,  
  codArtista varchar(255) not null,  
  constraint IDpartecipare_film primary key (codArtista, codFilm));
```

```
alter table partecipare_film add constraint FKpar_CAS  
  foreign key (codArtista)  
  references CAST;
```

```
alter table partecipare_film add constraint FKpar_FIL  
  foreign key (codFilm)  
  references FILM;
```

#### **- produrre ep**

```
create table produrre_ep (  
  codCasaProduttrice varchar(255) not null,  
  codEpisodio varchar(255) not null,  
  constraint IDprodurre_ep primary key (codCasaProduttrice, codEpisodio));
```

```
alter table produrre_ep add constraint FKpro_EPI  
  foreign key (codEpisodio)
```

```
references EPISODIO;
```

```
alter table produrre_ep add constraint FKpro_CAS  
foreign key (codCasaProduttrice)  
references CASA_PRODUTTRICE;
```

#### **- produrre\_film**

```
create table produrre_film (  
    codCasaProduttrice varchar(255) not null,  
    codFilm varchar(255) not null,  
    constraint IDprodurre_film primary key (codFilm, codCasaProduttrice));
```

```
alter table produrre_film add constraint FKpro_FIL  
foreign key (codFilm)  
references FILM;
```

```
alter table produrre_film add constraint FKpro_CAS  
foreign key (codCasaProduttrice)  
references CASA_PRODUTTRICE;
```

#### **- SERIE**

```
create table SERIE (  
    codiceSerie varchar(255) not null,  
    titolo varchar(255) not null,  
    durata time not null,  
    constraint IDSERIE primary key (codiceSerie));
```

#### **- STAGIONE**

```
create table STAGIONE (  
    codiceSerie varchar(255) not null,  
    numeroStagione int not null,  
    constraint IDSTAGIONI primary key (codiceSerie, numeroStagione));
```

```
alter table STAGIONE add constraint FKdividere  
foreign key (codiceSerie)
```

references SERIE;

### - **TITOLARE ACCOUNT**

```
create table TITOLARE_ACCOUNT (  
    CF varchar(255) not null,  
    nomeAccount varchar(255) not null,  
    indirizzo varchar(300) not null,  
    città varchar(255) not null,  
    numCartaDiCredito varchar(255) not null,  
    constraint IDTITOLARE_ACCOUNT primary key (CF),  
    constraint FKpossedere_ID unique (nomeAccount));
```

```
alter table TITOLARE_ACCOUNT add constraint FKpossedere_FK  
    foreign key (nomeAccount)  
    references ACCOUNT;
```

### - **UTENTE**

```
create table UTENTE (  
    nomeAccount varchar(255) not null,  
    nomeUtente varchar(255) not null,  
    constraint IDUTENTE primary key (nomeAccount, nomeUtente));
```

```
alter table UTENTE add constraint FKaccedere  
    foreign key (nomeAccount)  
    references ACCOUNT;
```

### - **VISIONE**

```
create table VISIONE (  
    nomeAccount varchar(255) not null,  
    nomeUtente varchar(255) not null,  
    data date not null,  
    ora time not null,  
    minutoArrivo time not null,  
    codEpisodio varchar(255),  
    codFilm varchar(255),  
    constraint IDVISIONE primary key (nomeAccount, nomeUtente, data, ora));
```

```
alter table VISIONE add constraint FKguardare  
foreign key (nomeAccount, nomeUtente)  
references UTENTE;
```

```
alter table VISIONE add constraint FKvisionare_ep  
foreign key (codEpisodio)  
references EPISODIO;
```

```
alter table VISIONE add constraint FKvisionare_f  
foreign key (codFilm)  
references FILM;
```

## 4.9 Traduzione delle operazioni in query SQL

### 4.9.1. Inserimento di un nuovo titolare account (con relativo account)

```
INSERT INTO ACCOUNT (nomeAccount, password)
VALUES (?, ?)
```

```
INSERT INTO TITOLARE_ACCOUNT (CF,nomeAccount,indirizzo,città,numCartaDiCredito)
VALUES (?, ?, ?, ?, ?)
```

### 4.9.2. Inserimento di un nuovo utente.

```
INSERT INTO UTENTE (nomeAccount,nomeUtente)
VALUES (?, ?)
```

### 4.9.3. Cancellazione di un titolare account esistente (con relativo account, abbonamenti e utenti)

```
DELETE FROM ABBONAMENTO
WHERE CF = ?
```

```
DELETE FROM TITOLARE_ACCOUNT
WHERE CF = ?
```

```
DELETE FROM ACCOUNT
WHERE nomeAccount = ?
```

```
DELETE FROM UTENTI
WHERE nomeAccount = ?
```

### 4.9.4. Cancellazione di un utente esistente.

```
DELETE FROM UTENTE
WHERE nomeUtente = ?
AND nomeAccount = ?
```

### 4.9.5. Numero di iscrizioni del mese corrente.

```
SELECT SUM() AS "Numero Iscrizioni"
FROM ABBONAMENTI
WHERE MONTH(dataInizio) = MONTH(NOW())
AND YEAR(dataInizio) = YEAR(NOW())
```

### 4.9.6 Ricerca per genere di una serie.

```
SELECT SERIE.*
FROM GENERE, appartenere_ep, EPISODIO, SERIE
WHERE GENERE.codGenere = ?
```

```
AND GENERE.codGenere = appartenere_ep.codGenere
AND EPISODIO.codEpisodio = appartenere_ep.codEpisodio
AND SERIE.codiceSerie = EPISODIO.codiceSerie
```

**4.9.7. Visualizzare il numero di stagioni e il numero di episodi per una data serie.**

```
SELECT SERIE.titolo, (SELECT COUNT(*)
                      FROM STAGIONE, SERIE
                      WHERE SERIE.titolo = ?
                      AND SERIE.codiceSerie = STAGIONE.codiceSerie) AS NumeroStagione,
COUNT(EPISODIO.codEpisodio) AS NumeroEpisodi
FROM EPISODIO, SERIE
WHERE SERIE.titolo = ?
AND SERIE.codiceSerie = EPISODIO.codiceSerie
GROUP BY SERIE.titolo
```

**4.9.8. Elenco degli utenti relativi ad un dato account.**

```
SELECT nomeUtente
FROM UTENTE
WHERE nomeAccount = ?
```

**4.9.9. Visualizzare la lista degli utenti.**

```
SELECT *
FROM UTENTI
```

**4.9.10. Individuare la serie TV più lunga presente sulla piattaforma.**

```
SELECT TOP(1) *
FROM SERIE
ORDER BY durata DESC
```

**4.9.11. Calcolare l'incasso dell'anno corrente.**

```
SELECT SUM(importo) AS IncassoAnnuo
FROM PAGAMENTO
WHERE YEAR(data) = YEAR (NOW())
```

**4.9.12. Individuare l'orario nel quale vengono viste più produzioni durante la giornata.**

```
SELECT TOP (1) ora
FROM VISIONE
GROUP BY ora
ORDER BY COUNT (*) DESC
```

#### **4.9.13. Migliori (più viste) serie TV nella piattaforma.**

```
SELECT TOP (10) SERIE.titolo, COUNT(*)/(SELECT COUNT(*)  
                                     FROM EPISODIO AS E  
                                     WHERE E.codiceSerie = SERIE.codiceSerie) AS NumeroVisioni  
FROM SERIE, EPISODIO, VISIONE  
WHERE SERIE.codiceSerie = EPISODIO.codiceSerie  
AND EPISODIO.codEpisodio = VISIONE.codEpisodio  
GROUP BY SERIE.codiceSerie, SERIE.titolo  
ORDER BY NumeroVisioni DESC;
```

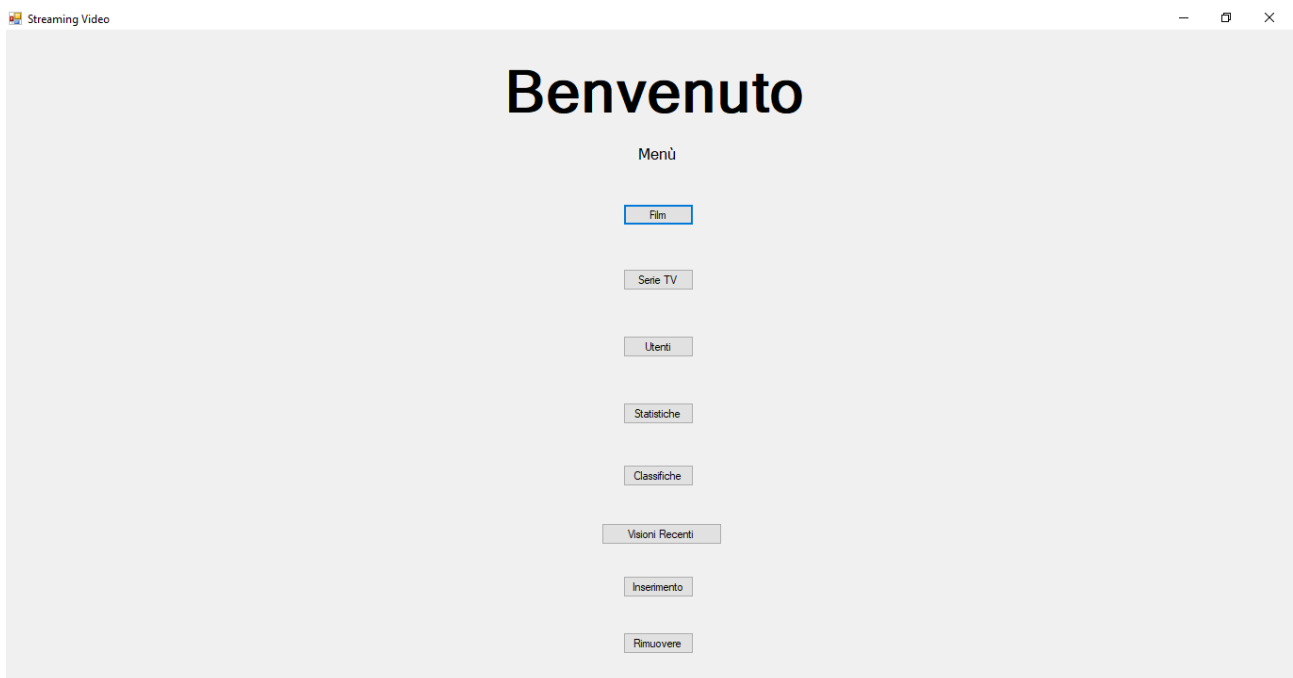
#### **4.9.14. Elenco delle ultime visioni effettuate.**

```
SELECT *  
FROM VISIONE  
WHERE data = GETDATE();
```

# 5. Applicazione

## 5.1 Descrizione dell'architettura dell'applicazione realizzata

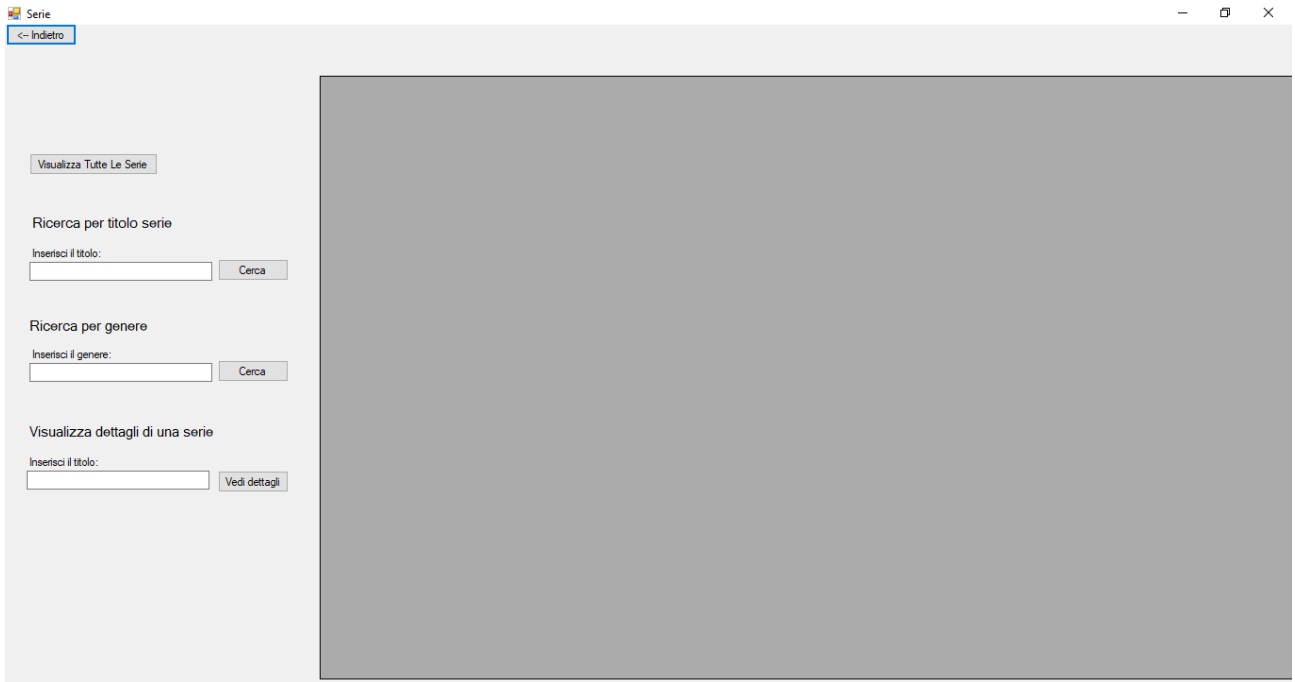
Si sviluppa un'applicazione per la gestione del database in linguaggio C# servendosi di LINQ, che renda possibile la messa in pratica delle operazioni. Il DB risiede in locale e usa SQL Server come DBMS. Si è fatto uso della classe LINQ TO SQL per mappare le tabelle del DB e rendere semplice lo sviluppo dell'applicazione. All'avvio viene proposta una schermata iniziale, ossia un menù destinato esclusivamente all'admin della piattaforma, che gli permette di svolgere tutte le operazioni.



**Figura 1.1** - Schermata principale dell'applicazione da cui accedere alle varie schermate



Cliccando i vari bottoni del menù vengono aperte schermate che permettono di accedere a dati più specifici per ogni argomento.



**Figura 1.2-** *Schermata di gestione delle Serie TV*

L'applicazione fornisce nel complesso le funzionalità richieste evidenziate nella fase di progettazione, includendo ulteriormente alcune banali operazioni non elencate.