

# ***PROGETTO BASI DI DATI***

*Anno Scolastico: 2019/2020*

Progetto realizzato da:

GAGLIOSTRO Letizia - Matricola: 896679

ARDUINO Marna - Matricola: 892690

# ***INDICE***

## **1. *Progettazione concettuale***

- 1.1. Requisiti Iniziali
- 1.2. Glossario dei termini
- 1.3. Requisiti rivisti e strutturati in gruppi di frasi omogenee
- 1.4. Schema ER + business rules

## **2. *Progettazione logica***

- 2.1. Tavola dei volumi
- 2.2. Tavola delle operazioni
- 2.3. Analisi delle ridondanze
- 2.4. Ristrutturazione schema ER
- 2.5. Schema ER Ristrutturato + business rules
- 2.6. Schema relazionale

## **3. *Implementazione***

- 3.1. DDL creazione database
- 3.2. DML di popolamento di tutte le tabelle del database
- 3.3. Operazioni di modifica e cancellazione

## Progettazione concettuale

### Requisiti iniziali

Si vuole realizzare una base di dati per la gestione di una piattaforma che fornisce informazioni sui film in uscita nei cinema, su serie e programmi TV in onda ispirato a piattaforme come ComingSoon.

L'applicazione prevede due tipi di utenti: utenti della redazione e utenti generici. Un numero limitato di utenti della redazione si occupa di aggiornare le informazioni relativi ai contenuti disponibili inserendo i dati di film e serie, comprese le date di uscita e programmazione in sala e TV.

I visitatori del sito che si sono registrati possono lasciare un voto (da 1 a 5 stelline) su qualunque contenuto. Inoltre gli utenti possono cliccare su un link "inserisci tra i preferiti" presente tra le schede in modo da visualizzare in seguito i contenuti salvati. Quando un utente si cancella dal sito i suoi preferiti possono essere rimossi mentre i voti rimangono.

Degli utenti si memorizzano nome utente, password e indirizzo email.

Il sito offre informazioni su una serie di contenuti: film, serie TV e programmi TV.

[Trama e Cast](#) [Video](#) [Foto](#) [Recensione](#) [News](#) [Streaming](#)

A row of six portrait photos of the main cast members of the film 'The Wailing' (2019). From left to right: Hye-jin Jang (KIM CHUNG-SOOK), Kang-ho Song (KIM KI-TAEK), Lee Sun-kyun (PARK DONG-IK), Cho Yeo-Jeong (PARK YEON-KYO), Choi Woo-sik (KIM KI-WOO), and Park So-dam (KIM KI-JUNG).

**Data di uscita:** 07 novembre 2019

**Genere:** Drammatico

**Anno:** 2019

**Regia:** Bong Joon Ho

**Attori:** Hye-jin Jang, Kang-ho Song, Lee Sun-kyun, Cho Yeo-Jeong, Choi Woo-sik, Park So-dam

**Paese:** Corea del Sud

**Durata:** 132 min

**Distribuzione:** Academy Two

**Sceneggiatura:** Kim Dae-hwan, Bong Joon Ho, Jin Won Han

**Fotografia:** Hong Kyung-pyo

**Musiche:** Jaeil Jung

**Produzione:** Barunson E&A, CJ E&M Film Financing & Investment Entertainment & Comics, CJ Entertainment, Frontier Works Comic



Genere: [Drammatico](#)

Ideatore: [Vince Gilligan](#)

Attori: [Bryan Cranston](#), [Anna Gunn](#), [Aaron Paul](#), [RJ Mitte](#), [Dean Norris](#), [Betsy Brandt](#), [Bob Odenkirk](#), [Jonathan Banks](#), [Giancarlo Esposito](#)

Anno: [2008](#)

Paese: [USA](#)

Produzione: [Sony Pictures Television](#)

Durata: [47 min](#)

*schema proposto dal sito comingsoon per le Serie TV, è stato aggiunto siccome è stato possibile ispirarsi a questo sito.*

Genere: [Musicale](#)

Anno: [1951](#)

Paese: [Italia](#)

Durata: [240 min](#)

Stato: [In Lavorazione](#)

*Schema proposto sul sito comingsoon per quanto riguarda i programmi TV.*

Programmi e serie TV vanno in onda su diverse piattaforme (es. Netflix, Rai Gulp,...) che possono essere canali TV tradizionali o piattaforme di streaming video.

Le serie TV sono organizzate per genere, come i film (esempio: commedia, drammatico...). Le serie TV sono composti da episodi, racchiusi in una o più stagioni, dove ogni episodio ha uno specifico titolo, durata, regista, e sceneggiatore. Anche per i programmi sono memorizzate alcune informazioni, come l'anno di messa in onda, genere, paese, durata e una descrizione.

I film vengono proiettati in cinema di tutto il paese. Per i cinema si vuole memorizzare il nome, contatti e localizzazione (regione, provincia, indirizzo).

Le proiezioni hanno orari specifici e diverse date, sono proiettati in una delle sale del cinema, comporta il pagamento di un biglietto indicato sul sito.

Scheda di esempio per un episodio di una serie TV.

### Stagione 1/5

Frustrato dal lavoro e dalla situazione familiare - un impiego nella scuola pubblica come professore di chimica, una moglie incinta e un figlio affetto da paralisi cerebrale - l'ormai cinquantenne Walter White decide di dare una svolta alla sua vita, soprattutto quando scopre di avere il cancro e sente la responsabilità di garantire un futuro alla sua famiglia, anche quando non sarà più con i suoi affetti. Considerate le sue conoscenze, Walter inizia a produrre metamfetamina, ma ha bisogno di qualcuno che gli crei un giro di clienti, cosa che lui non sarebbe in grado di fare, sia perché non ha le conoscenze giuste, sia perché gli manca la "faccia tosta". Si affida quindi al "talento" di Jesse Pinkman, un suo ex studente. I due, però, scoprono ben presto che produrre illegalmente della droga non è facile come venderla, soprattutto se hai un parente che lavora per l'antidroga e un manipolo di concorrenti che non vogliono farsi rovinare gli affari.

#### Curiosità

Inizialmente la prima stagione prevedeva nove episodi. Tuttavia, lo sciopero degli sceneggiatori americani limitò la produzione a sette puntate.

La prima stagione ha avuto su AMC una media d'ascolto di 1.3 milioni di spettatori.

### Lista episodi Stagione 1

#### 1. Questione di chimica (Pilot)



**Regia:** Vince Gilligan

**Sceneggiatura:** Vince Gilligan

**Durata:** 47 min

Ad ogni film e serie TV è associata una lista di attori, con l'indicazione del personaggio interpretato.

Alcune informazioni tipiche dei film, come la regia, per le serie TV sono specifiche dei singoli episodi.

Gli attori e registi hanno una scheda con le seguenti caratteristiche:



## Ezio Greggio biografia

### DATI ANAGRAFICI DI EZIO GREGGIO

**Età:** 65 anni

**Nasce a:** Cossato, Biella (Italia)

**Nasce il:** 07/04/1954

### BIOGRAFIA DI EZIO GREGGIO

Attore. Nasce nel 1954 a Cossato, Biella (Italia) (ha 65 anni).

Tra i suoi film come regista, interprete, sceneggiatore, ricordiamo:

[Super Vacanze di Natale](#) (2017), [Box office 3D: Il film dei film](#) (2011), [Occhio a quei due](#) (2009), [Il papà di Giovanna](#) (2008), [Un'estate al mare](#) (2008), [Un maresciallo in gondola](#) (2002), [2001 - Un'astronave spuntata nello spazio](#) (2000), [Svitati](#) (1999), [Anni '60](#) (1999), [Anni '50](#) (1998), [Killer per caso](#) (1997), [Dracula morto e contento](#) (1995), [Selvaggi](#) (1995), [Miracolo Italiano](#) (1994), [Il silenzio dei prosciutti](#) (1993), [Anni 90](#) (1992), [Infelici e contenti](#) (1992), [Vacanze di Natale '91](#) (1991), [Occhio alla perestrojka](#) (1990), [Vacanze di Natale '90](#) (1990), [Montecarlo Gran Casinò](#) (1987), [Yuppies, i giovani di successo](#) (1986), [Italian Fast Food](#) (1986), [SBAMM!](#) (1980),

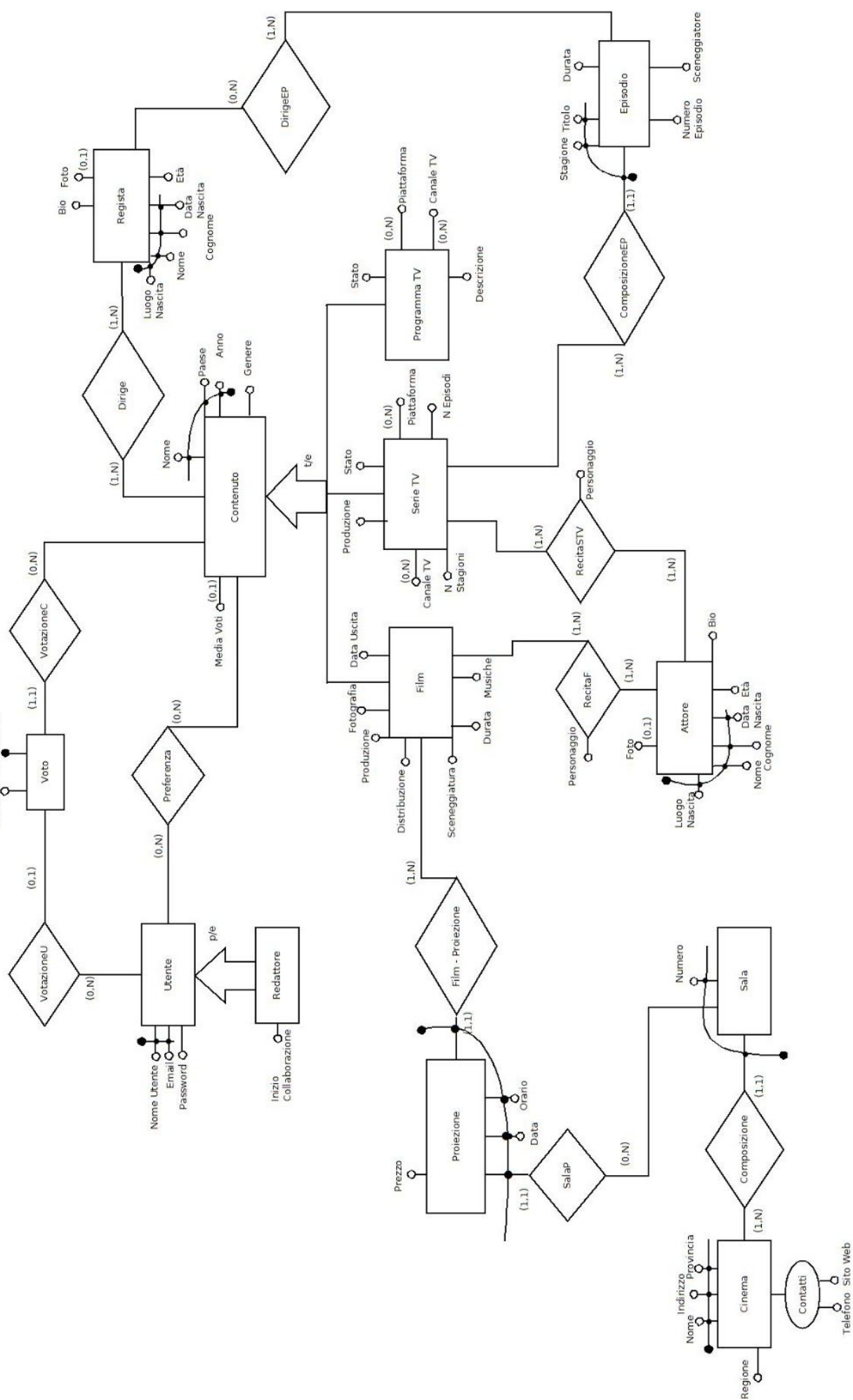
Può essere anche mostrata una lista dei contenuti più recenti a cui la persona ha partecipato.

## Glossario dei termini

| Termini    | Descrizione  | Sinonimi             | Collegamenti         |
|------------|--|----------------------|----------------------|
| Utente     | Coloro che visitano e interagiscono con i contenuti del sito. Possono essere registrati e/o redattori. | Visitatore           | Contenuto            |
| Contenuto  | Contenuti del sito. Possono essere serie TV, film e programmi TV.                                      | Spettacolo           | Utente,Voto          |
| Proiezione | Film proiettato in un cinema.  | Film                 | Contenuto,Cinema     |
| Regista    | Una persona che ha diretto la creazione di un contenuto.   | Regia                | Contenuto            |
| Attore     | Una persona che ha recitato in uno spettacolo.   |                      | Contenuto            |
| Cinema     | Luogo dove vengono proiettati i film.  | Sala cinematografica | Contenuto,Proiezione |
| Voto       | Voti lasciati sui contenuti dagli utenti registrati.   | Valutazione          | Contenuto, Utente    |



N stelline Cardinalità





## **Business rules**

- Un utente se è redattore può modificare le informazioni sui contenuti inserendo dati di film e serie TV, date di uscita e programmazione in sala e in TV.
- Il Numero di stelline dei voti dev'essere un numero compreso tra 1 e 5.
- Quando un utente viene eliminato allora vengono eliminate anche le sue preferenze, invece i voti devono rimanere.
- La media dei voti dev'essere calcolata come la media matematica dei voti su quel contenuto che si sta considerando. La cardinalità è 0 se un contenuto non ha nessun voto.
- I contenuti recenti di attori e registi devono coincidere con i contenuti a cui hanno preso parte.
- Il numero di episodi di una Serie TV deve coincidere con il numero di associazioni con l'entità Episodio.
- L'attributo Stagione dell'entità Episodio di una serie TV deve essere un numero minore o uguale all'attributo N Stagioni di Serie TV.
- Gli attributi Canale TV e Piattaforma di Serie TV sono a 0 se la serie Tv non viene più trasmessa.
- Gli attributi Canale TV e Piattaforma di Programmi sono a 0 se un programma non viene più trasmesso.

## **Commenti:**

Si considerano solamente utenti registrati, ovvero non sono considerati gli utenti che possono visitare il sito senza essere registrati tramite nome utente, email e password. Si è fatta questa scelta poichè in questo modo è più facile/intuitivo identificare gli utenti, altrimenti non sarebbe stato nè efficiente nè semplice individuare ed identificare anche utenti che non hanno nessun elemento distintivo ed identificativo come nome utente e email.

Per i contatti dell'entità Cinema si suppone che ogni cinema metta a disposizione il suo contatto telefonico e il sito web.

Inoltre per i voti si è deciso di aggiungere un attributo che serve per indicare la cardinalità dei voti in modo da rendere identificabile ogni singolo voto. L'attributo quando viene inserito il primo voto nel sito vale 1, man mano che si aggiungono voti viene fatto aumentare.

Inoltre per gli attributi di serie TV, film e programmi ci siamo ispirati all'organizzazione del sito ComingSoon.

I contenuti recenti di un attore o regista verranno mostrati tramite una query nell'ultima fase del progetto.

## ***Progettazione logica***

**Tavola dei volumi**

| <b>Concetto</b> | <b>Tipo</b> | <b>Volume</b> |
|-----------------|-------------|---------------|
| Cinema          | E           | 1500          |
| Sala            | E           | 3000          |
| Film            | E           | 200.000       |
| Proiezione      | E           | 1500          |
| Attore          | E           | 13.000        |
| Serie TV        | E           | 50.000        |
| Episodio        | E           | 1.000.000     |
| Programma TV    | E           | 50.000        |
| Contenuto       | E           | 300.000       |
| Regista         | E           | 300.000       |
| Utente          | E           | 1.000.000     |
| Redattore       | E           | 100           |
| Voto            | E           | 4.000.000     |
| Composizione    | A           | 3000          |
| Film-Proiezione | A           | 1500          |
| RecitaF         | A           | 39.000        |
| RecitaSTV       | A           | 52.000        |
| ComposizioneEP  | A           | 1.000.000     |
| DirigeEP        | A           | 300.000       |
| Dirige          | A           | 450.000       |
| VotazioneC      | A           | 4.000.000     |

|            |   |            |
|------------|---|------------|
| VotazioneU | A | 4.000.000  |
| Preferenza | A | 10.000.000 |
| SalaP      | A | 3000       |

A= associazione , E = entità

Commenti:

Il numero di cinema è stato deciso dopo numerose ricerche ed è emerso che in Italia ci sono più di 1000 cinema e più di 3000 sale, quindi i numeri scelti sono una stima di questi dati.

Approssimativamente possiamo dire che esistono più di 1.000.000.000 di film nel mondo se si tengono conto anche di quelli poco diffusi in occidente (es quelli di Bollywood e del resto dell'Asia) e senza contare i film che usciranno in futuro, ma stimiamo che il database possa contenere circa 200.000 film.

Secondo il sito ComingSoon in Italia ci sono più di 1000 proiezioni, perciò si sceglie un numero vicino a quest'ultimo per rappresentare la realtà il più possibile.

In media si stima di avere circa 20 episodi per serie TV, quindi  $100.000 * 20$  si ha circa 1.000.000 episodi tra tutte le serie TV.

Per i contenuti si è deciso di fare una somma del numero di serie TV, programmi e film (perciò  $200.000 + 50.000 + 50.000 = 300.000$ ).

Si suppone che in media per ogni film, serie e programma TV ci sia un regista anche se alcuni contenuti sono presenti più registi.

Gli utenti totali supponiamo siano 1.000.000, mentre i redattori sono solo una piccola parte quindi saranno circa 100.

Supponendo che su 1.000.000 di utenti solo circa 200.000 lascino dei voti su 20 contenuti.

Perciò si arriva a concludere :  $200.000 * 20 = 4.000.000$ .

Per gli attori si suppone che in media lavorino a 6 film (quindi  $13.000 * 3 = 39.000$ ) e a 8 serie TV (quindi  $13.000 * 4 = 52.000$  ).

Si suppone che in media un regista diriga almeno 10 episodi per ogni serie TV, quindi  $30.000 * 10 = 300.000$ .

Si suppone che un regista nel corso della sua vita diriga almeno 15 contenuti tra serie Tv, programmi e film, quindi  $30.000 * 15 = 450.000$ .

Per gli utenti si suppone che possano salvare nella lista dei preferiti un massimo di 10 contenuti, perciò si ottiene che in totale il database possa contenere 10.000.000 preferenze.

## Tavola delle operazioni

| Operazione | Descrizione  | Tipo | Frequenza            |
|------------|--|------|----------------------|
| 1          | Registrazione utente.  | I    | 50 alla settimana    |
| 2          | Login utente.  | I    | 50 alla settimana    |
| 3          | Login redattore.   | I    | 2 alla settimana     |
| 4          | Aggiornamento* informazioni film.  | I    | 1 alla settimana     |
| 5          | Aggiornamento* informazioni serie TV.                                      | I    | 1 alla settimana     |
| 6          | Aggiornamento* informazioni programmi TV.                                  | I    | 1 alla settimana     |
| 7          | Aggiornamento programmazione al cinema e in TV.                            | I    | 1 alla settimana     |
| 8          | Aggiornamento date di uscita di film, programmi e serie TV                 | I    | 1 alla settimana     |
| 9          | Ricerca informazioni relative ad un contenuto.                             | I    | 1000 volte al giorno |
| 10         | Ricerca informazioni relative alle proiezioni.                             | I    | 1000 volte al giorno |
| 11         | Ricerca informazioni relative ad un attore.                                | I    | 50 volte al giorno   |
| 12         | Ricerca informazioni relative ad un regista.                               | I    | 20 volte al giorno   |
| 13         | Ricerca informazioni relative a un cinema                                  | I    | 100 volte al giorno  |
| 14         | Ricerca di informazioni relative a un contenuto sulla base della votazione | I    | 100 volte al giorno  |
| 15         | Rilascio di un voto su un contenuto.                                       | I    | 200 volte al giorno  |
| 16         | Cancellazione utente normale o redattore.                                  | I    | 2 a settimana        |
| 17         | Inserimento di un contenuto nella lista dei preferiti.                     | I    | 200 volte al giorno  |
| 18         | Memorizza un nuovo episodio relativo ad una serie TV                       | I    | 1 volta a settimana  |

|    |  |   |                    |
|----|--|---|--------------------|
| 19 | Per ogni attore trova la lista dei contenuti a cui ha preso parte.                                   | B | 2 a settimana      |
| 20 | Per ogni regista trova la lista dei contenuti che ha diretto.  | B | 2 a settimana      |
| 21 | Stampa i dati di una proiezione.   | B | 2 a settimana      |
| 22 | Per ogni cinema trova le proiezioni e le sale  | B | 1 a settimana      |
| 23 | Per ogni contenuto trova le informazioni relative a produzioni, durata, distribuzione ...            | B | 1 a settimana      |
| 24 | Per ogni serie TV si vuole stampare l'elenco degli episodi, il titolo di ogni episodio, le stagioni. | B | 10 volte al giorno |

#### **I: Interattiva**

#### **B: Batch**

\*si intende la rimozione, l'aggiunta oppure la modifica di contenuto.

#### **Commento:**

Le operazioni che vengono fatte più di frequente sono le ricerche.

La base di dati prevede diverse tipologie di ricerche: in base al contenuto (operazione 9), sulle proiezioni per verificare quali film sono trasmessi al cinema (operazione 10) e sui cinema dove l'utente può vedere se un certo film viene proiettato o meno (operazione 13). Inoltre si dà la possibilità agli utenti di cercare informazioni relative agli attori e registi (operazioni 11 e 12).

Le operazioni dalla 4 alla 8 e l'operazione 18 possono essere effettuate solo da utenti redattori, siccome prevedono la modifica dei contenuti del database.

Nelle operazioni batch troviamo tutte quelle operazioni che vengono lanciate e si lasciano eseguire e, se lo si desidera, è possibile memorizzare queste informazioni all'interno di files.

Tra le operazioni batch si sono scelte quelle operazioni che stampano tutte le informazioni relative a determinate entità come, ad esempio, la stampa di tutte le informazioni relative ad un contenuto.

## Analisi delle ridondanze

Ridondanze individuate:

- Numero di episodi di una serie TV: posso ricavarlo contando semplicemente il numero di associazioni di Composizione EP.
- Media dei voti di un contenuto: posso ricavarlo facendo la somma tra tutti i voti diviso il numero totale dei voti su quel contenuto (ricavabile dall'associazione Votazione C).

### Ridondanza su N Episodi



Il numero di episodi di una Serie TV è una ridondanza perché posso ricavarlo contando gli episodi associati tramite ComposizioneEP ad una serie TV.

### TAVOLA DEI VOLUMI

| Concetto       | Tipo | Volume    |
|----------------|------|-----------|
| Serie TV       | E    | 50.000    |
| Episodio       | E    | 1.000.000 |
| ComposizioneEP | A    | 1.000.000 |

### TAVOLA DELLE OPERAZIONI

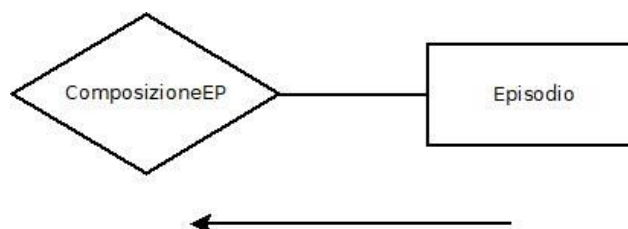
| Operazione | Descrizione  | Tipo | Frequenza             |
|------------|--|------|-----------------------|
| 1          | Memorizza un nuovo episodio relativo ad una serie TV.  | I    | 1 volta a settimana*. |
| 2          | Per ogni serie TV si vuole stampare l'elenco degli episodi, il titolo di ogni episodio, le stagioni. | B    | 10 al giorno.         |

\*si suppone che in media per le serie TV esca un episodio a settimana

## Scenario A: assenza di ridondanza

### OPERAZIONE 1

Schema di operazione



Lo schema di operazione non coinvolge serie TV perché non devo aggiornare le informazioni relative ad una serie TV.

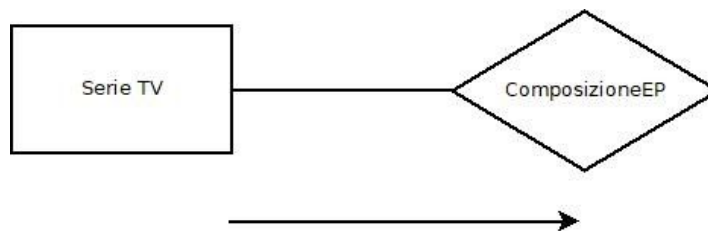
La freccia indica il verso di lettura.

*Tavola degli accessi*

| Concetto       | Costrutto    | Accessi | Tipo      |
|----------------|--------------|---------|-----------|
| Episodio       | Entità       | 1       | Scrittura |
| ComposizioneEP | Associazione | 1       | Scrittura |

## OPERAZIONE 2

Schema di operazione



Non ho l'entità N Episodi quindi bisogna contare gli episodi usando l'associazione ComposizioneEP.

*Tavola degli accessi*

| Concetto       | Costrutto    | Accessi | Tipo    |
|----------------|--------------|---------|---------|
| Serie TV       | Entità       | 1       | Lettura |
| ComposizioneEP | Associazione | 20      | Lettura |

Viene effettuato l'accesso all'entità *Serie TV* per leggere le informazioni relative alle serie TV. Nella tavola volumi abbiamo 50.000 serie TV e 1.000.000 di occorrenze di *ComposizioneEP*, quindi per una singola serie TV abbiamo in media  $1.000.000/50.000 = 20$  accessi a ComposizioneEP.



## Scenario B: con ridondanza

### OPERAZIONE 1

Schema di operazione

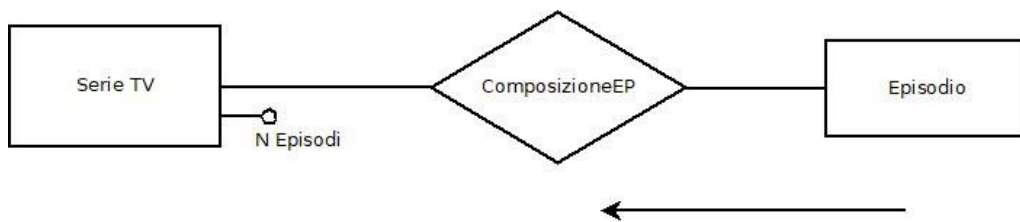


Tavola degli accessi

| Concetto       | Costrutto    | Accessi | Tipo      |
|----------------|--------------|---------|-----------|
| Episodio       | Entità       | 1       | Scrittura |
| ComposizioneEP | Associazione | 1       | Scrittura |
| Serie TV       | Entità       | 1       | Lettura   |
| Serie TV       | Entità       | 1       | Scrittura |

Quando aggiungo un nuovo episodio bisogna leggere il valore vecchio del numero di episodi e successivamente incrementarlo, questo spiega i due accessi (uno in lettura e uno in scrittura) sull'entità *Serie TV*.

### OPERAZIONE 2

Schema di operazione



Tutte le informazioni relative ad una serie TV le ho sull'entità *Serie TV*, non ho bisogno di accedere in lettura o in scrittura ad altri costrutti.

Tavola degli accessi

| Concetto | Costrutto | Accessi | Tipo    |
|----------|-----------|---------|---------|
| Serie TV | Entità    | 1       | Lettura |

### Costi scenario A

- **Spazio occupato dalla ridondanza:** 0 byte.
- **Tempo**
  - Operazione 1: posso trascurare gli accessi al giorno perché l'operazione viene fatta una volta a settimana.
  - Operazione 2: 20 (accessi in lettura) \* 10 (accessi al giorno) = 200 accessi al giorno.

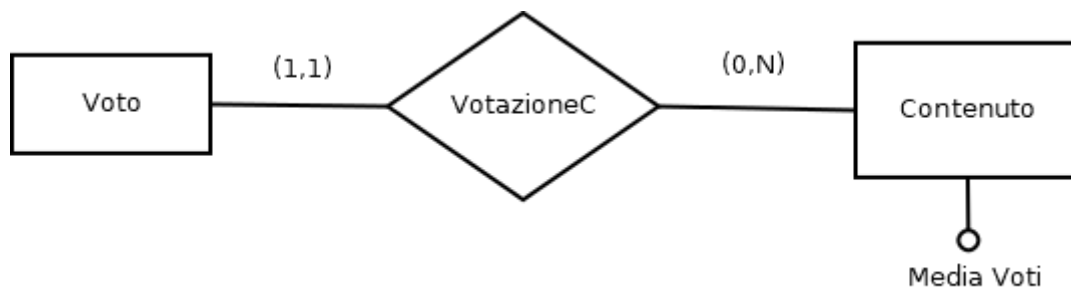
In totale ho **200 accessi al giorno**.

### Costi scenario B

- **Spazio occupato dalla ridondanza:** rappresento il numero di episodi con 2 byte. Nel database prevedo 2 byte per ogni serie TV:  $50.000 * 2 = 100.000$  byte = 100 kB.  
Con  $2^{28}$  riesco a rappresentare  $2^{16}$  diversi valori, va bene perché è possibile che una serie abbia più di 300 episodi quindi non rappresentabili con 1 byte.
- **Tempo**
  - Operazione 1: posso trascurare gli accessi perché l'operazione viene fatta una volta a settimana.
  - Operazione 2: ho un accesso in lettura, il costo è basso quindi può essere trascurabile.

**Conclusione:** si è deciso di conservare la ridondanza per poter risparmiare 200 accessi al giorno al costo di aggiungere 100 kB per memorizzare il numero di episodi.

### Ridondanza su Media dei voti di un contenuto



Media dei voti di un contenuto è una ridondanza perché è possibile ricavare la media facendo la somma tra tutti i voti diviso il numero totale dei voti su quel contenuto (ricavabile dall'associazione *VotazioneC*).

### TAVOLA DEI VOLUMI

| Concetto   | Tipo | Volume    |
|------------|------|-----------|
| Contenuto  | E    | 300.000   |
| Voto       | E    | 4.000.000 |
| VotazioneC | A    | 4.000.000 |

### TAVOLA DELLE OPERAZIONI

| Operazione | Descrizione                          | Tipo | Frequenza           |
|------------|--------------------------------------|------|---------------------|
| 1          | Rilascio di un voto su un contenuto. | I    | 200 volte al giorno |
| 2          | Stampa i dati relativi al contenuto  | B    | 1 volta a settimana |

## Scenario A: assenza di ridondanza

### OPERAZIONE 1

Schema di operazione

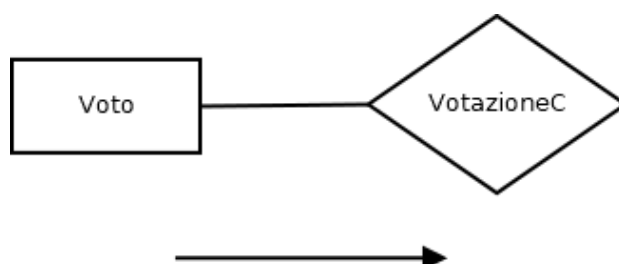


Tavola degli accessi

| Concetto   | Costrutto    | Accessi | Tipo      |
|------------|--------------|---------|-----------|
| Voto       | Entità       | 1       | Scrittura |
| VotazioneC | Associazione | 1       | Scrittura |

Il rilascio di un voto su un contenuto non richiede il coinvolgimento dell'entità contenuto siccome non viene modificata nessuna informazione relativa a quest'ultimo.

### OPERAZIONE 2

Schema di operazione

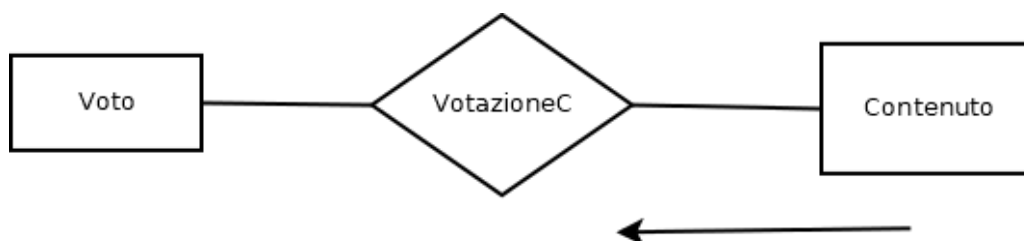


Tavola degli accessi

| Concetto   | Costrutto    | Accessi | Tipo    |
|------------|--------------|---------|---------|
| Contenuto  | Entità       | 1       | Lettura |
| VotazioneC | Associazione | 14      | Lettura |
| Voto       | Entità       | 14      | Lettura |

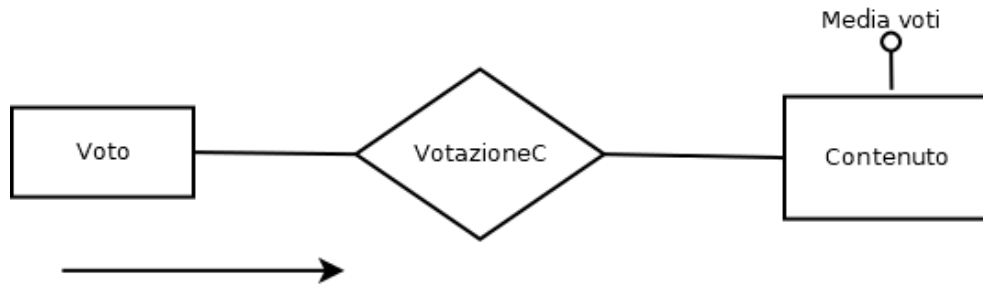
Per calcolare la media dei voti relativa ad un contenuto devo accedere sia all'entità *VotazioneC* sia all'entità *Voto*, perchè l'associazione aiuta a determinare quanti voti ha un particolare contenuto mentre l'entità *Voti* fornisce il numero di stelline.

Nella tavola volumi abbiamo 300.000 contenuti e 4.000.000 di occorrenze di *VotazioneC*, quindi per una singolo contenuto abbiamo in media  $4.000.000/300.000 = 13,33$  (arrotondato per eccesso a 14) accessi a *VotazioneC*.

## Scenario B: con ridondanza

### OPERAZIONE 1

#### Schema di operazione



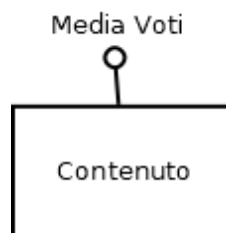
#### Tavola degli accessi

| Concetto   | Costrutto    | Accessi | Tipo      |
|------------|--------------|---------|-----------|
| Voto       | Entità       | 1       | Scrittura |
| VotazioneC | Associazione | 1       | Scrittura |
| Contenuto  | Entità       | 1       | Lettura   |
| Contenuto  | Entità       | 1       | Scrittura |

Quando il contenuto riceve un nuovo voto bisogna leggere prima la vecchia media dei voti e successivamente aggiornarla, questo giustifica i due accessi (uno in lettura e uno in scrittura) sull'entità *Contenuto*.

### OPERAZIONE 2

#### Schema di operazione



Tutte le informazioni relative a un contenuto le ho sull'entità *Contenuto*, non ho bisogno di accedere in lettura o in scrittura ad altri costrutti.

#### Tavola degli accessi

| Concetto  | Costrutto | Accessi | Tipo    |
|-----------|-----------|---------|---------|
| Contenuto | Entità    | 1       | Lettura |

### Costi scenario A

- **Spazio occupato dalla ridondanza:** 0 byte.
- **Tempo**
  - Operazione 1: 200 accessi al giorno \* (2\*2) = 800 accessi al giorno.
    - E' stato raddoppiato il valore perché si tratta di un'operazione di scrittura che ha costo maggiore.
  - Operazione 2: posso trascurare gli accessi al giorno perché l'operazione viene fatta una volta a settimana.

In totale ho **800 accessi al giorno**.

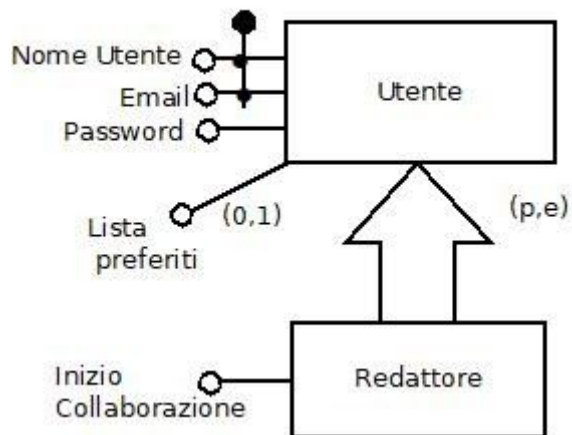
### Costi scenario B

- **Spazio occupato dalla ridondanza:** rappresento la media dei voti con 1 byte, con 1 byte riesco a rappresentare  $2^{18}$  numeri. In media ho 14 voti quindi mi servono 14 byte.
- **Tempo**
  - Operazione 1: (200 accessi al giorno \* 3(accessi in scrittura))\*2 = 1200 accessi al giorno.
    - il valore è stato raddoppiato perché gli accessi in scrittura costano di più di quelli in lettura.
  - Operazione 2: posso trascurare gli accessi perché l'operazione viene fatta una volta a settimana.

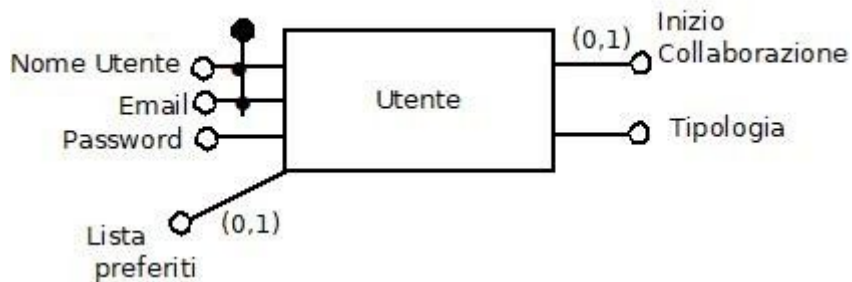
**Conclusione:** Si è deciso di togliere la ridondanza a fronte di risparmiare  $1200-800 = 400$  accessi al giorno.

## Ristrutturazione schema ER

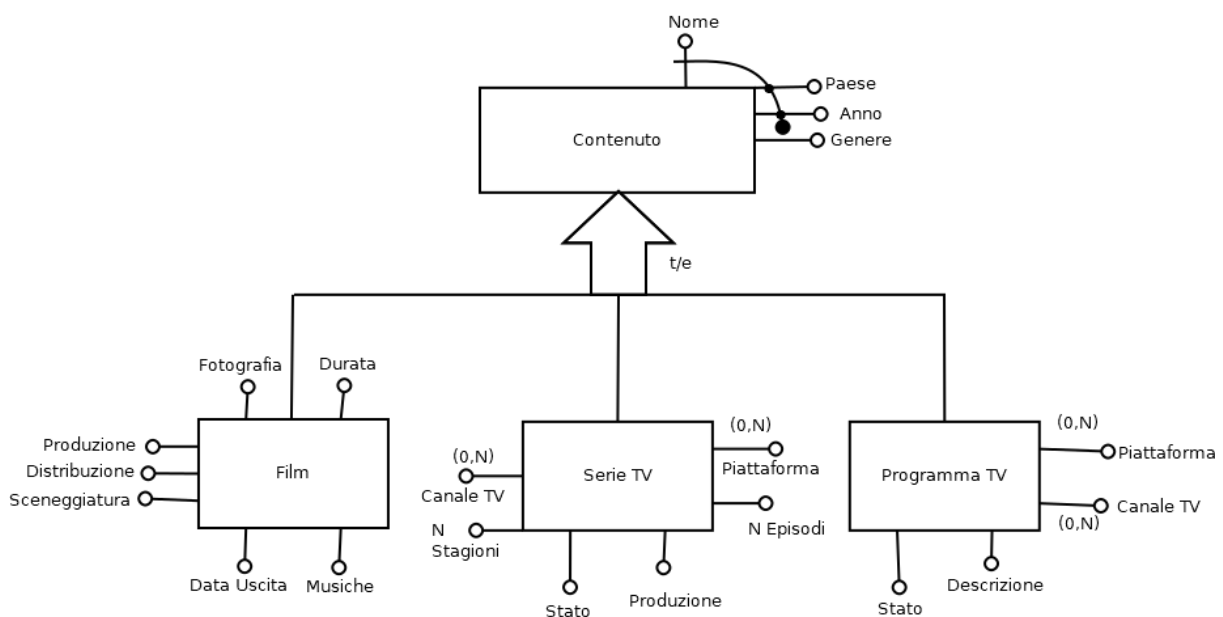
### Eliminazione delle generalizzazioni



Modifica proposta:

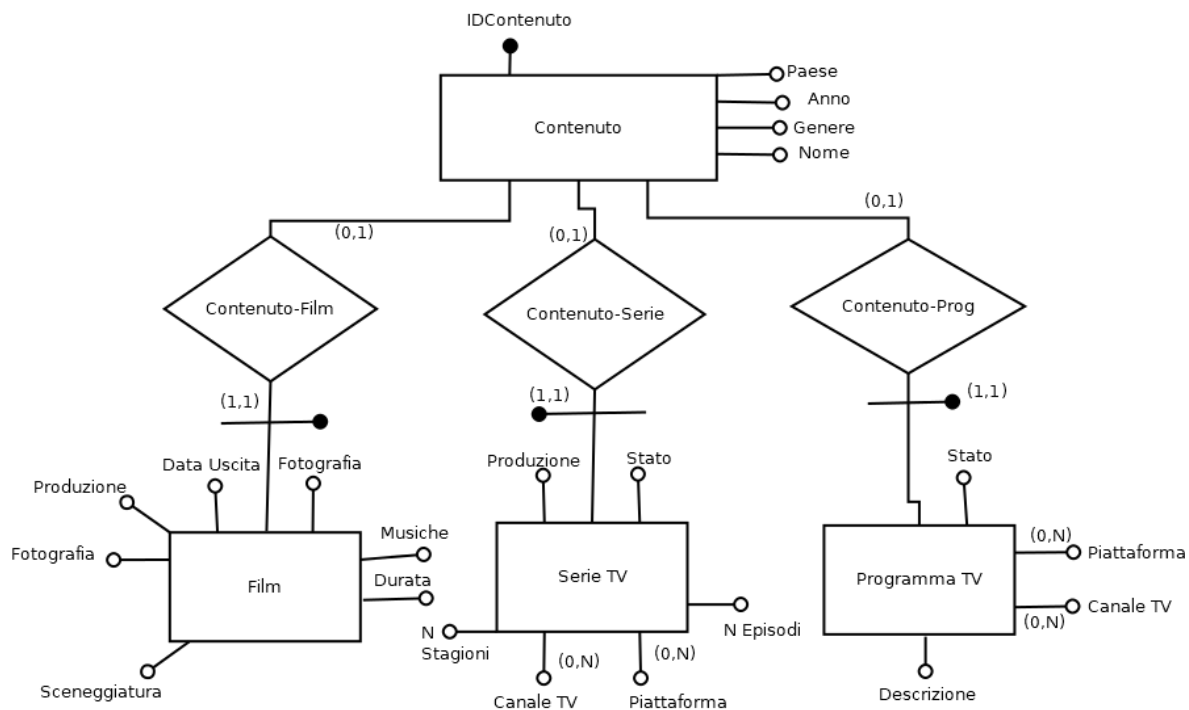


La generalizzazione viene eliminata accorpendo l'entità *Redattore* con *Utente* e vengono aggiunti due attributi: *Tipologia* indica di quale tipo di utente si tratta (redattore oppure utente generico), invece *Inizio Collaborazione* rappresenta la data di inizio collaborazione di un *Redattore*. L'attributo è 0 quando si tratta di un utente generico, mentre è 1 quando si tratta di un Redattore.





Modifica proposta:



Nella base di dati si fa molta distinzione tra *Contenuto* e tra *Film*, *Serie TV* e *Programma TV* perciò si è deciso di mantenere i figli separati dall'entità genitore.

Ogni occorrenza di *Contenuto* non può partecipare contemporaneamente alle associazioni *Film-Contenuto*, *SerieTV-Contenuto*, *ProgTv-Contenuto*. In altre parole se una cardinalità è a 1 le altre sono a 0 e non ci può essere il caso in cui le cardinalità sono tutte a 0.

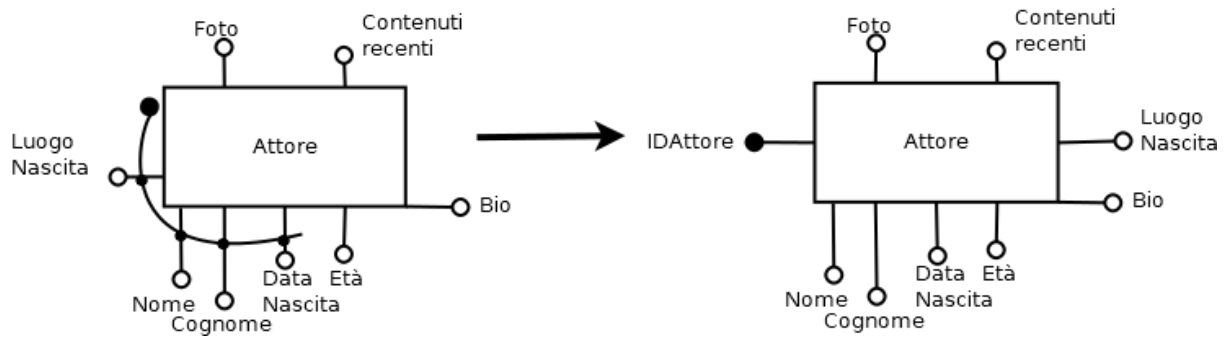
### Identificatori principali

Alcune Entità nessun identificatore rispetta i criteri di *assenza di opzionalità*, *semplicità e utilizzo delle operazioni più frequenti/importanti*, si è deciso di introdurre nuovi attributi contenenti valori speciali generati appositamente per questo scopo.

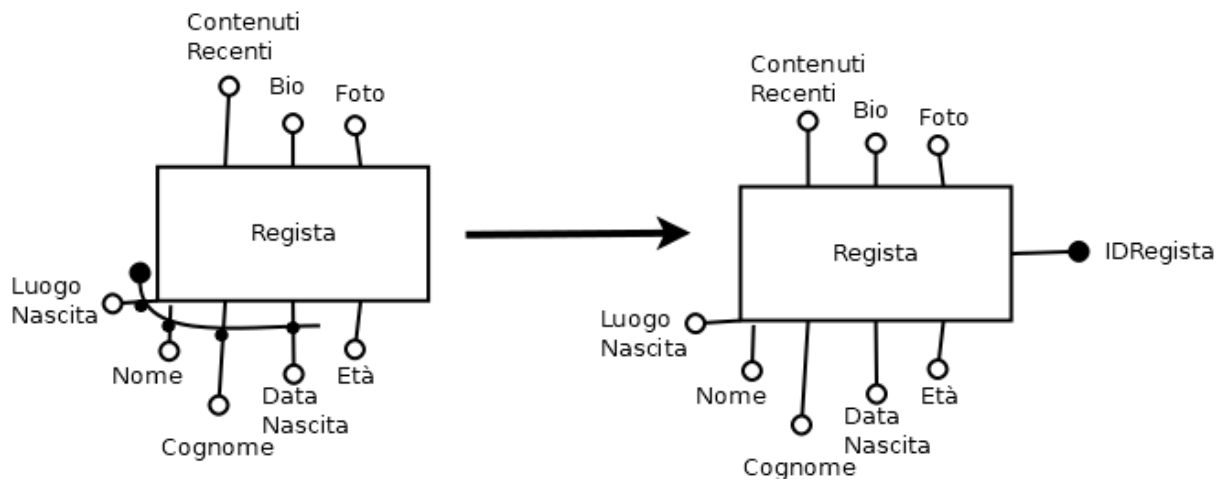
Si sono introdotti gli identificatori solo per le entità che verranno maggiormente usate all'interno del database in modo da ottimizzare le operazioni più frequenti, in questo caso sono: Attori, Registi, Contenuti.

Per le altre entità si è deciso di lasciare gli identificatori come erano stati definiti nello schema concettuale.

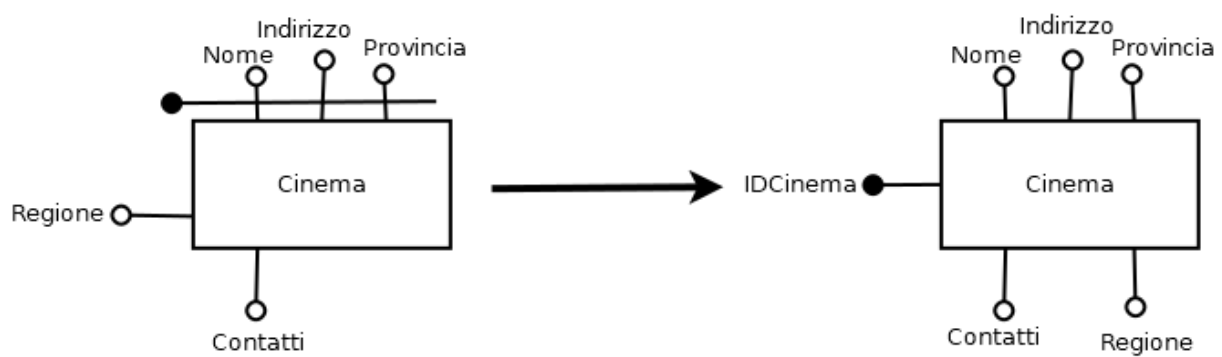
### Entità Attori:



### Entità Registi:

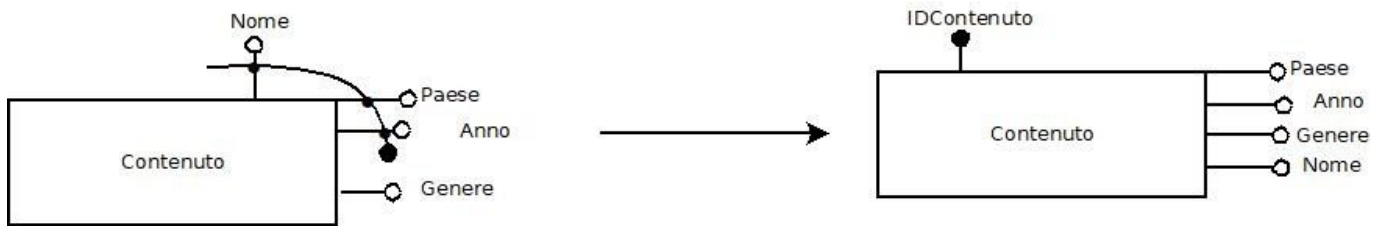


### Entità Cinema:



### Entità contenuto:

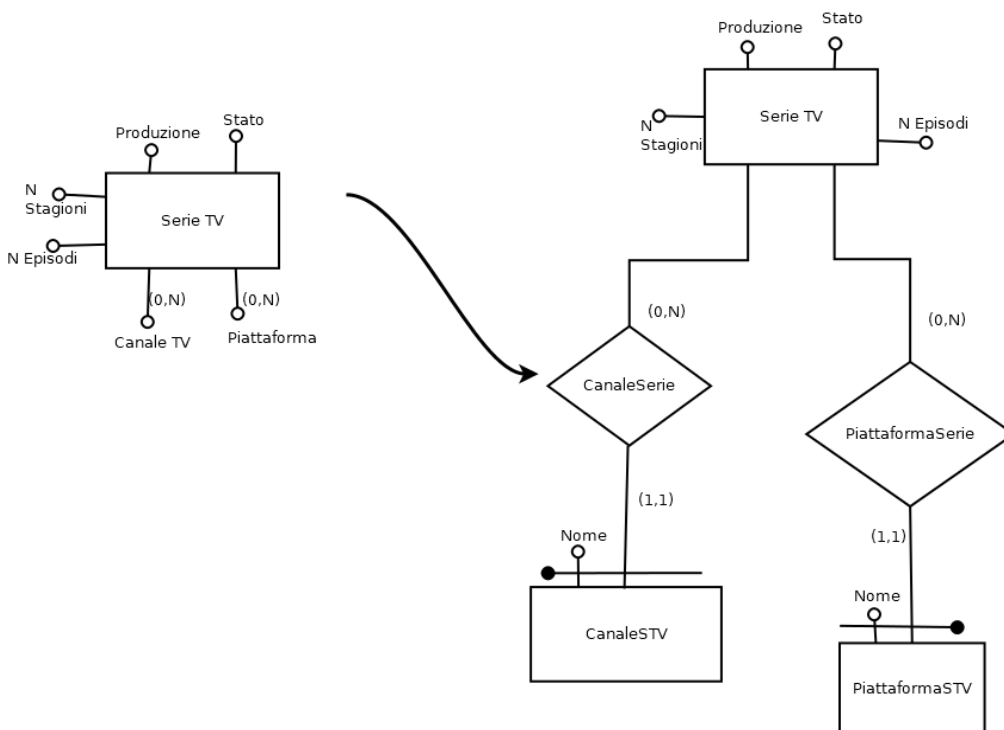
Siccome è un'entità in cui ci sono molti accessi si è deciso di introdurre un unico ID per identificare in maniera più semplice e immediata un contenuto.



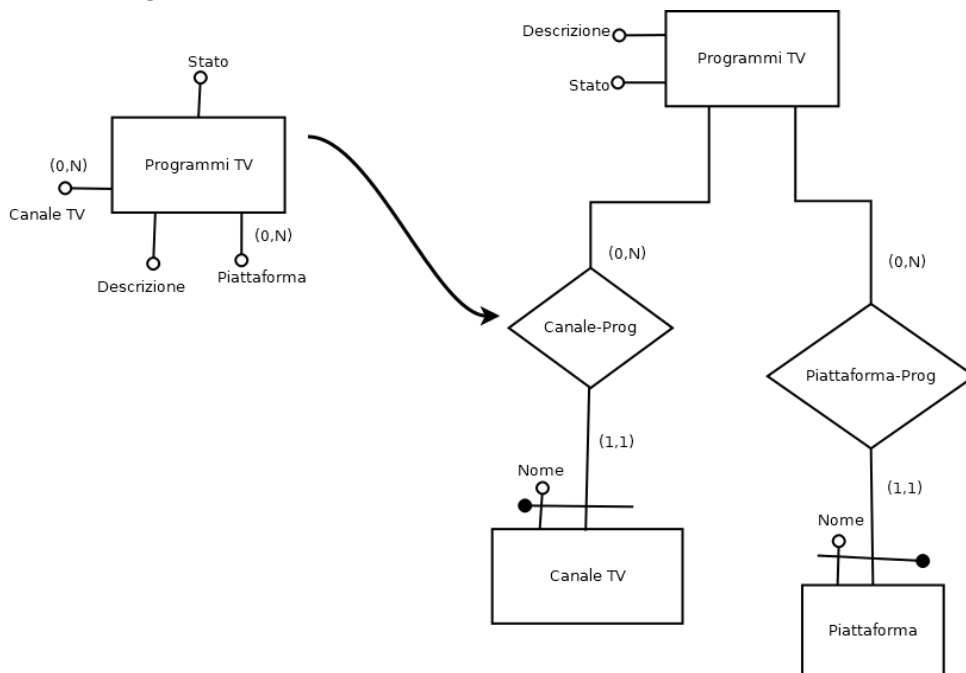
### Trasformazione attributi multivalore

Il modello relazionale non prevede l'uso di attributi multivalore, perciò vanno modificati.

### Entità serie TV



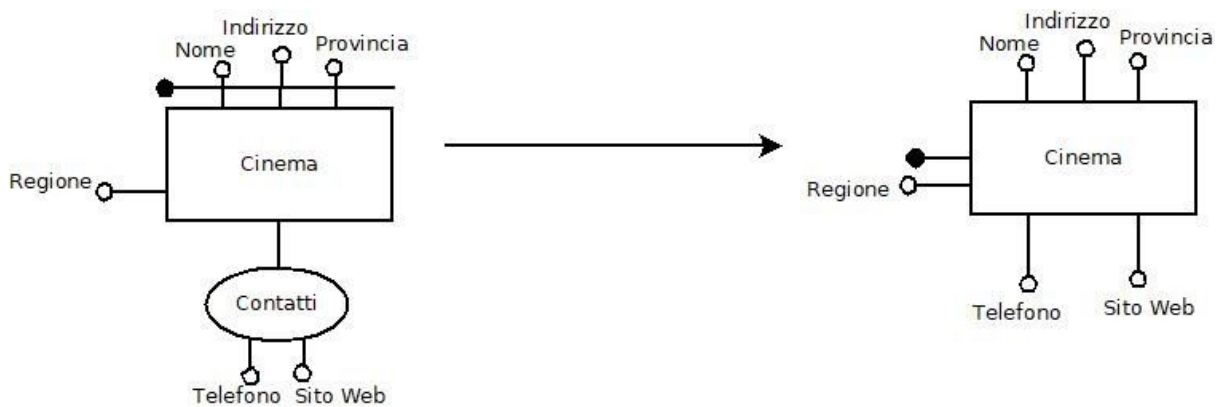
## Entità Programma TV

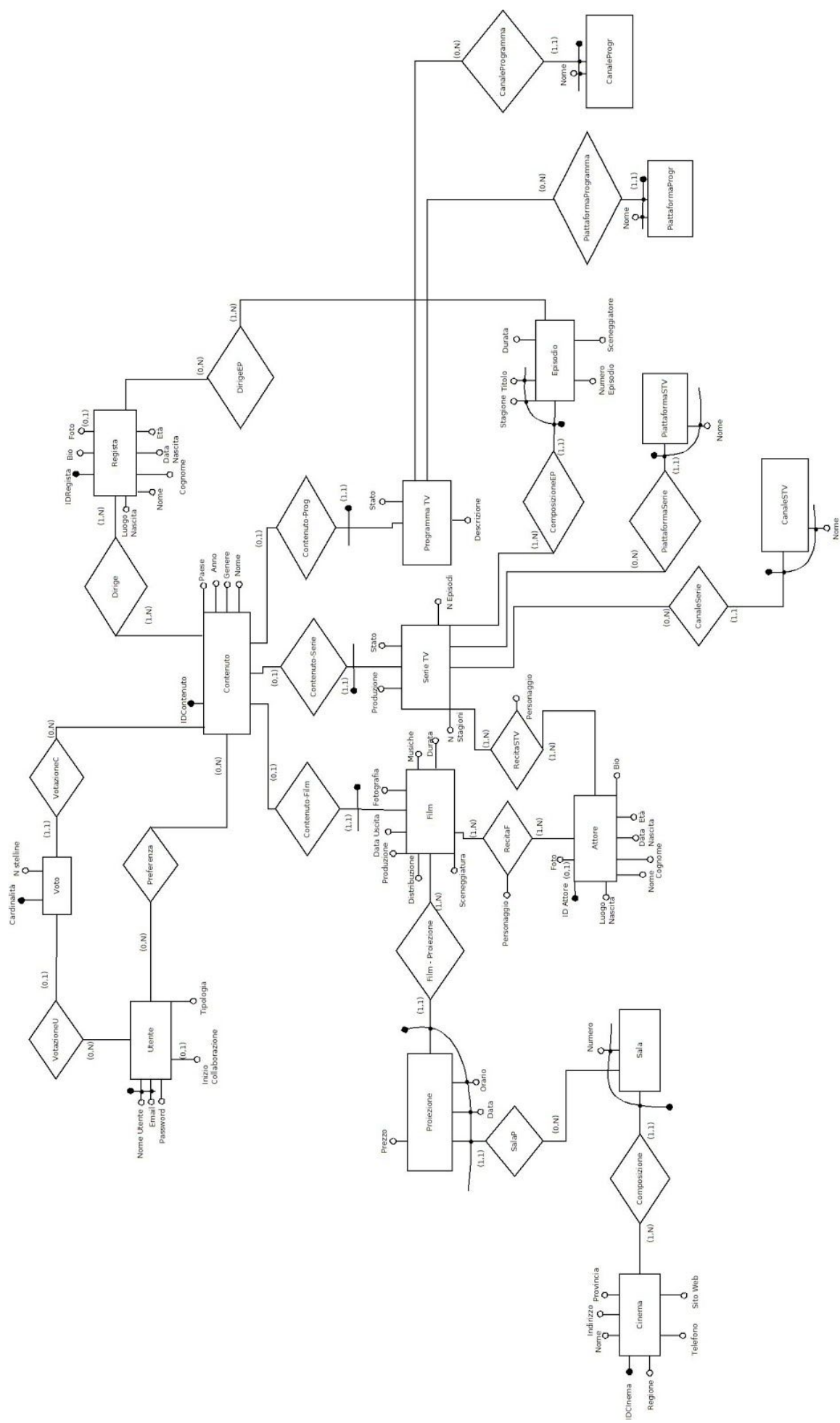


Vengono aggiunte due nuove entità per rappresentare i canali TV e le piattaforme.

## Eliminazione attributi composti

Gli attributi composti non possono essere rappresentati direttamente nel modello relazionale quindi devono essere trasformati.

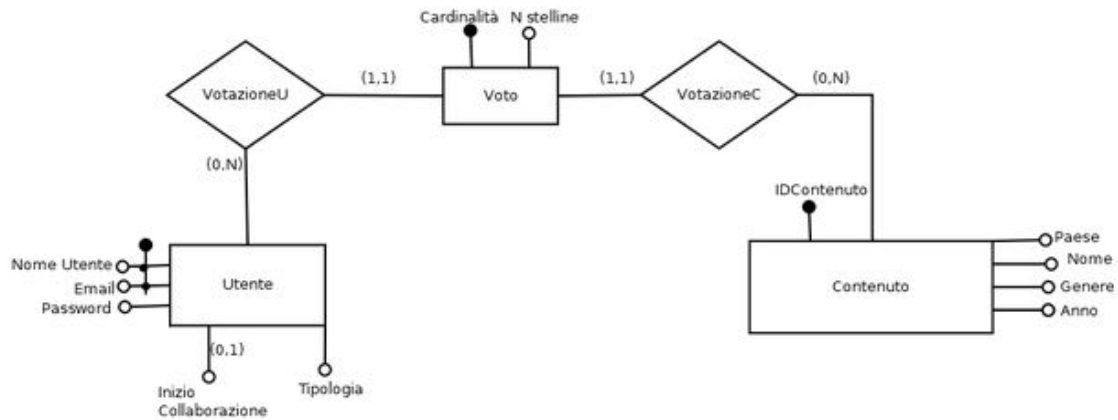




## Business rules

- L'attributo Tipologia dell'entità Utente può essere "Generico" oppure "Redattore".
- Se l'attributo Tipologia dell'entità Utente ha valore "Redattore" allora è possibile inserire un valore in Inizio Collaborazione,
- Il numero di stelline dei voti dev'essere un numero compreso tra 1 e 5.
- Quando un utente viene eliminato allora vengono eliminate anche le sue preferenze, i voti invece devono rimanere.
- L'attributo IDContenuto cambia a seconda se si fa riferimento a una serie TV, un film oppure una serie TV. In caso di film si ha ID formato dalla lettera F seguita da dei numeri, per le serie TV una lettera S seguita da dei numeri e per i programmi una lettera P seguita da dei numeri.
- Il numero di episodi di una Serie TV deve coincidere con il numero di associazioni con l'entità Episodio.
- L'attributo Stagione dell'entità Episodio di una serie TV deve essere un numero minore o uguale all'attributo N Stagioni di Serie TV.
- Le associazioni che collegano ProgrammaTV a CanaleProg e PiattaformaProg hanno cardinalità 0 se i programmi non vengono più trasmessi in televisione oppure non sono presenti in nessuna piattaforma.
- Le associazioni che collegano SerieTV a CanaleSTV e PiattaformaSTV hanno cardinalità 0 se le Serie TV non vengono più trasmesse in televisione oppure non sono presenti in nessuna piattaforma.
- Le cardinalità delle associazioni che vanno da Contenuto a Film, SerieTV, ProgrammaTV non possono essere tutte a 1 contemporaneamente. In altre parole, un contenuto può essere solo un Film o una Serie TV o un Programma TV ma non due o tutte e tre le cose contemporaneamente.

## Schema relazionale



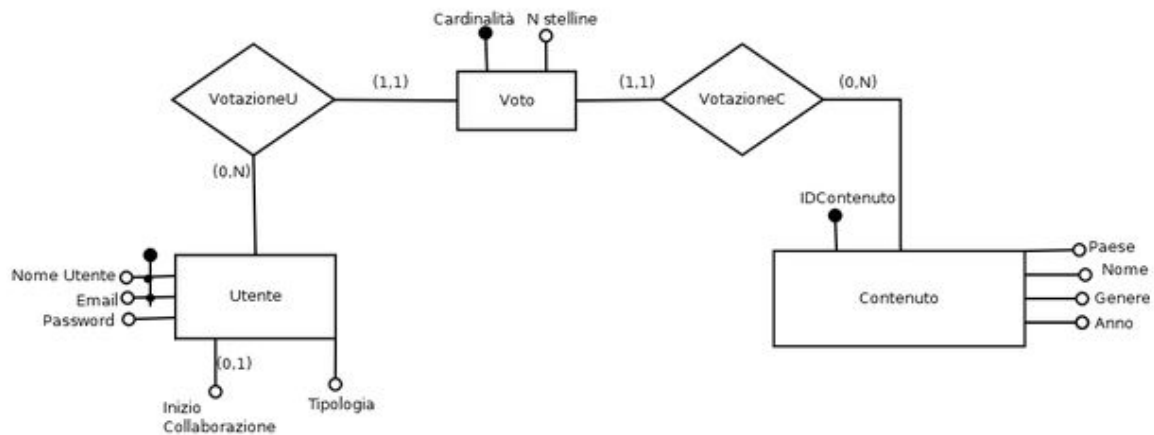
*Utente*(NomeUtente,Email, Password, InizioCollaborazione\*, Tipologia)

*Contenuto*(IDContenuto, Nome, Paese, Genere, Anno)

*Voto*(Cardinalità, NumeroStelline, *Contenuto*Utente\*, EmailUtente\*)

*Voto*(Utente, EmailUtente) referencia *Utente*(NomeUtente, Email)

*Voto*(Contenuto) referencia *Contenuto*(IDContenuto)



*Utente*(NomeUtente,Email, Password, ListaPreferiti\*, InizioCollaborazione\*, Tipologia)

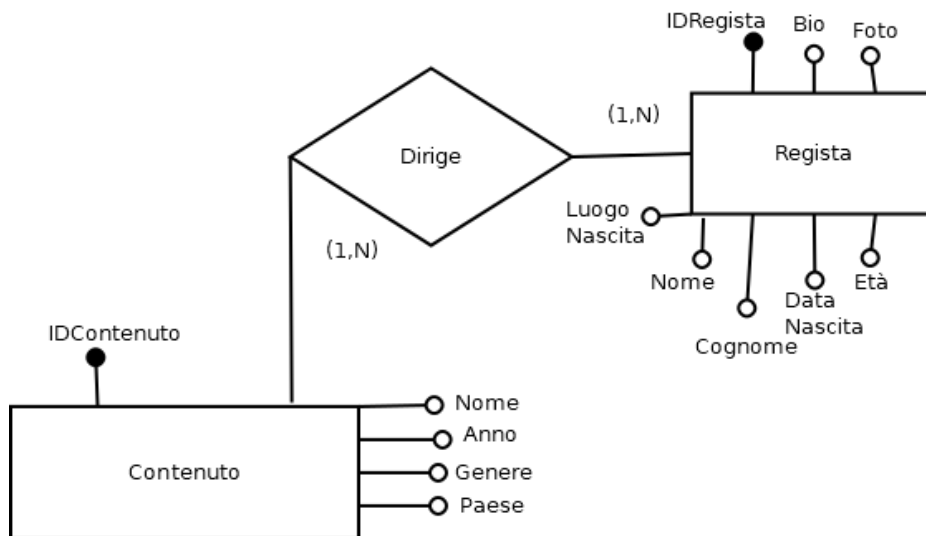
*Contenuto*(IDContenuto, Nome, Paese, Genere, Anno )

*Preferenza*(EmailUtente, *Utente*, *Contenuto*)

*Preferenza*(EmailUtente, *Utente*) referencia *Utente*(Email, NomeUtente)

*Preferenza*(Contenuto) referencia *Contenuto*(IDContenuto)





*Contenuto*(IDContenuto, Nome, Paese, Genere, Anno )

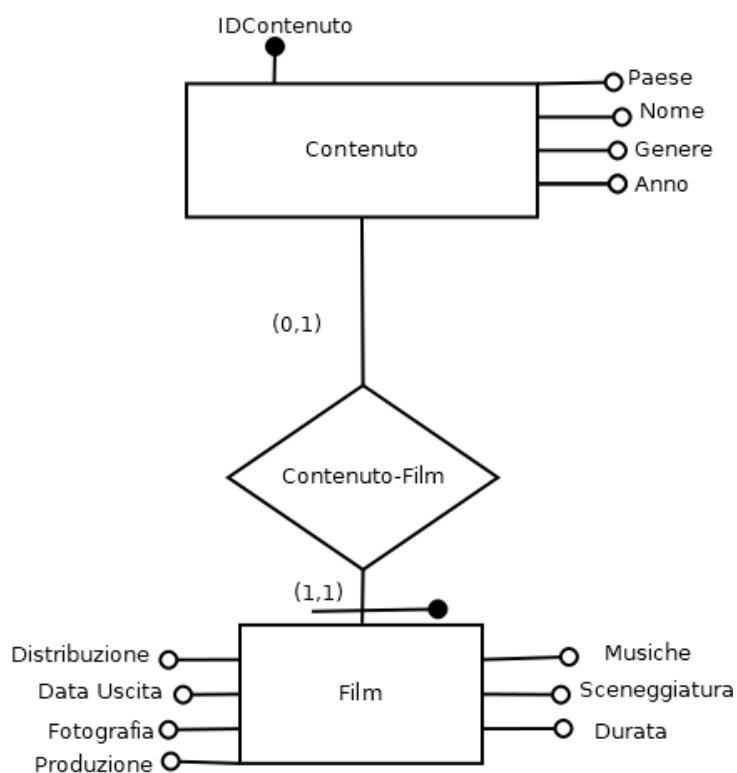
*Regista*(IDRegista, Bio, Foto\*, Nome, Cognome, DataNascita, Età, LuogoNascita)

*Dirige*(Contenuto, Regista)

*Dirige*(Contenuto) referencia *Contenuto*(IDContenuto)

*Dirige*(Regista) referencia *Regista*(IDRegista)

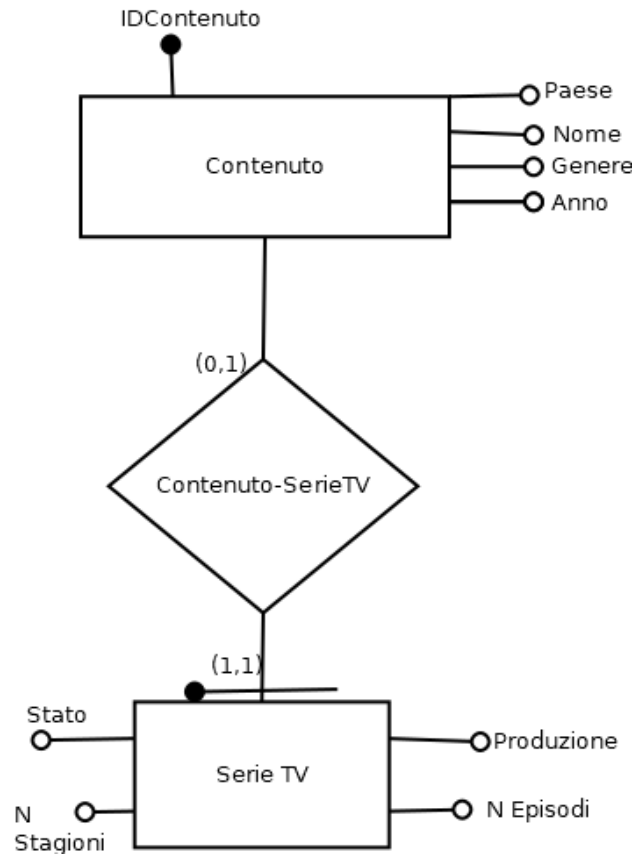
chiave: IDContenuto, Nome



*Contenuto*(IDContenuto, Nome, Paese, Genere, Anno)

*Film*(CodiceFilm, Distribuzione, Musica, Produzione, DataUscita, Sceneggiatura, Fotografia, Durata)

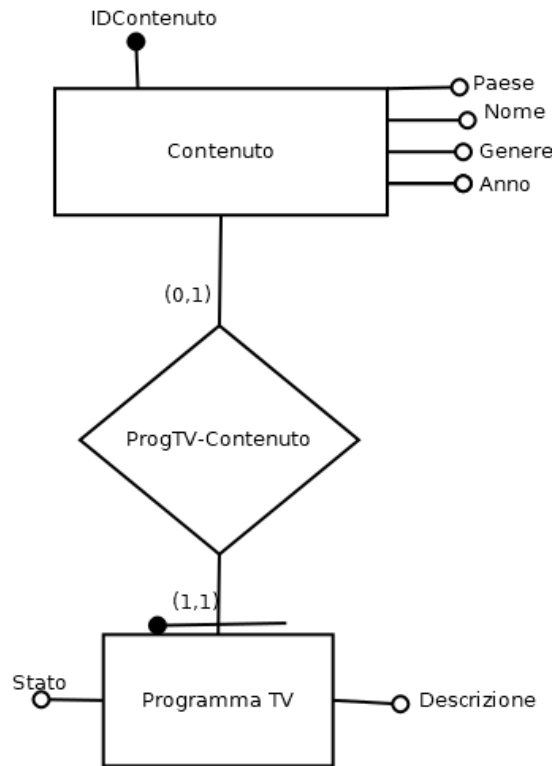
*Film*(CodiceFilm) referencia *Contenuto*(IDContenuto)



*Contenuto*(IDContenuto, Nome, Paese, Genere, Anno)

*SerieTV*(CodiceSerie, NStagioni, NEpisodi, Produzione, Stato)

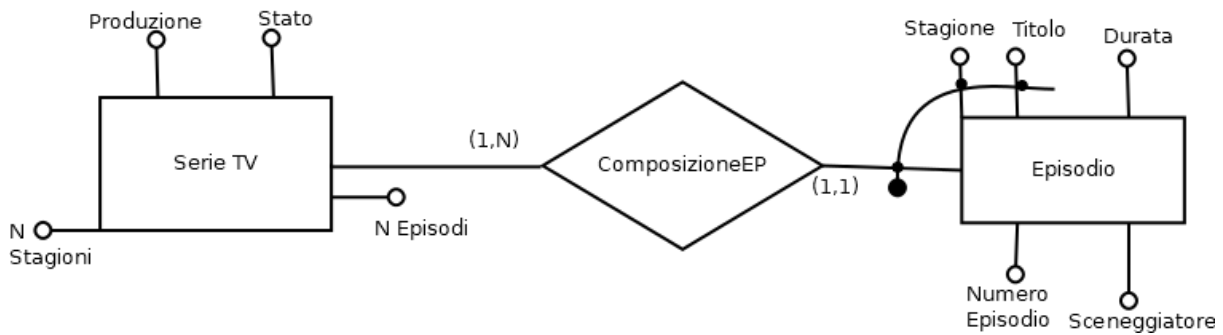
*SerieTV*(CodiceSerie) referencia *Contenuto*(IDContenuto)



*Contenuto*(IDContenuto, Nome, Paese, Genere, Anno)

*ProgrammaTV*(CodiceProgramma, Descrizione, Stato)

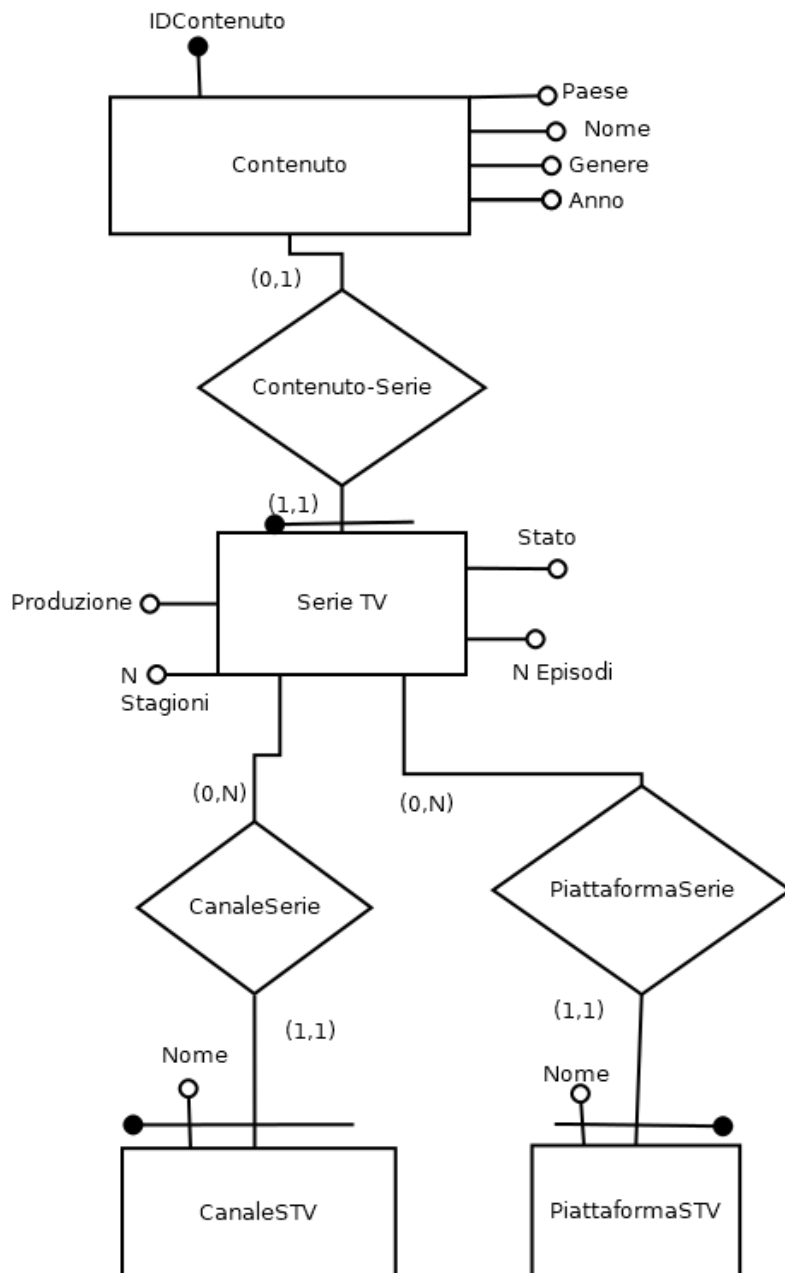
*ProgrammaTV*(CodiceProgramma) referencia *Contenuto*(IDContenuto)



*Episodio*(Stagione, Titolo, SerieTV, Durata, NumeroEpisodio, Sceneggiatore)

*SerieTV*(CodiceSerie, NStagioni, NEpisodi, Produzione, Stato)

*Episodio*(SerieTV) referencia *SerieTV*(CodiceSerie)



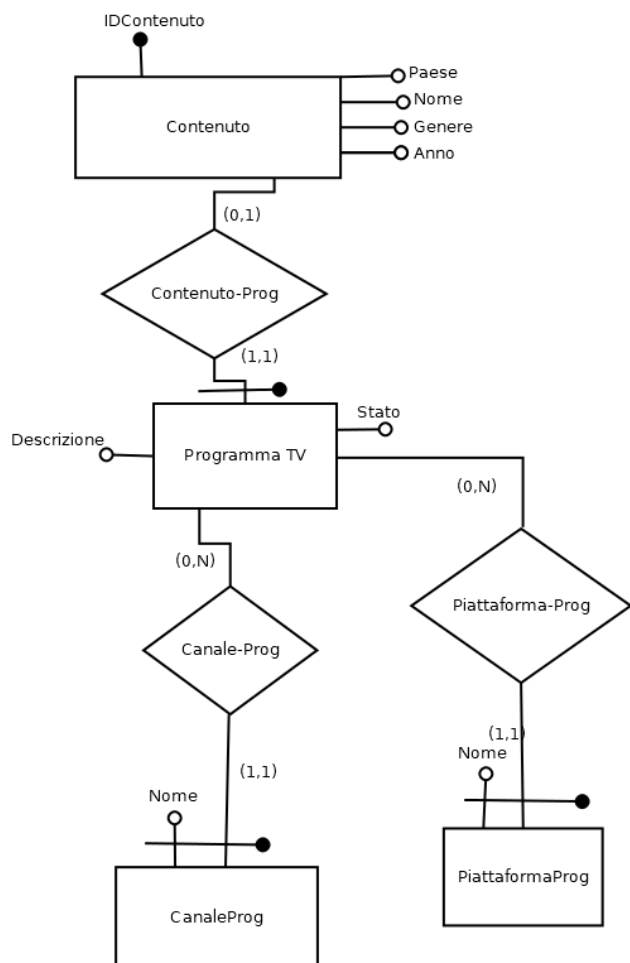
*CanaleSTV*(Nome,*SerieTV*)

*PiattaformaSTV*(Nome,*SerieTV*)

*SerieTV*(CodiceSerie,*NStagioni*,*NEpisodi*,*Produzione*,*Stato*)

*CanaleSTV*(*SerieTV*) *referenzia* *SerieTV*(*CodiceSerie*)

*PiattaformaSTV*(*SerieTV*) *referenzia* *SerieTV*(*CodiceSerie*)



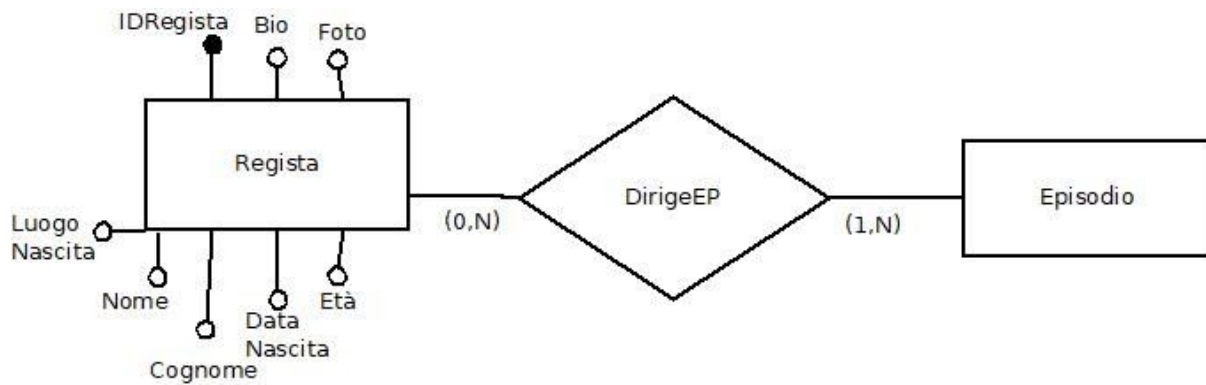
*CanaleProg*(Nome,ProgrammaTV)

*PiattaformaProg*(Nome,ProgrammaTV)

*ProgrammaTV*(CodiceProgramma,Descrizione,Stato)

*CanaleProg*(ProgrammaTV) referencia *ProgrammaTV*(CodiceProgramma)

*PiattaformaProg*(ProgrammaTV) referencia *ProgrammaTV*(CodiceProgramma)



*Sono stati omessi gli attributi e gli identificatori di Episodio per rendere più semplice l'analisi*

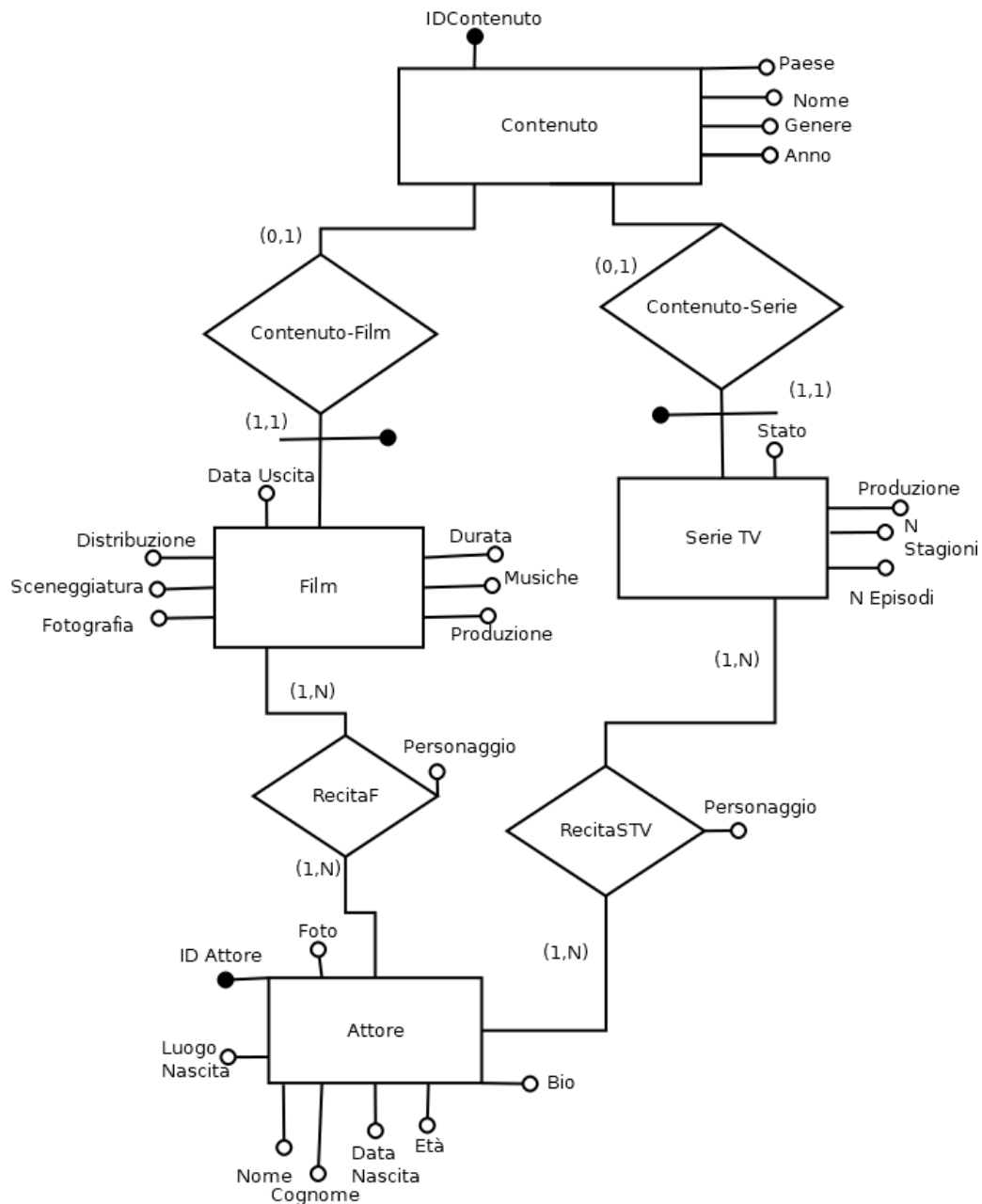
*Episodio*(Stagione, Titolo, SerieTV, Durata, NumeroEpisodio, Sceneggiatore)

*Regista*(IDRegista, Bio, Foto\*, Nome, Cognome, DataNascita, Età, Luogo Nascita)

*DirigeEP*(Stagione, Titolo, SerieTV, Regista)

*DirigeEP*(Stagione, SerieTV, Titolo) referencia *Episodio*(Stagione, SerieTV, Titolo)

*DirigeEP*(Regista) referencia *Regista*(IDRegista)



*Film*(CodiceFilm, Musiche, Distribuzione, Produzione, Data Uscita, Sceneggiatura, Fotografia, Durata)

*SerieTV*(CodiceSerie, NStagioni, NEpisodi, Produzione, Stato)

*Attore*(IDAttore, LuogoNascita, Foto\*, Nome, Cognome, DataNascita, Età, Bio)

*RecitaF*(Film, Attore, Personaggio)

*RecitaSTV*(SerieTV, Attore, Personaggio)

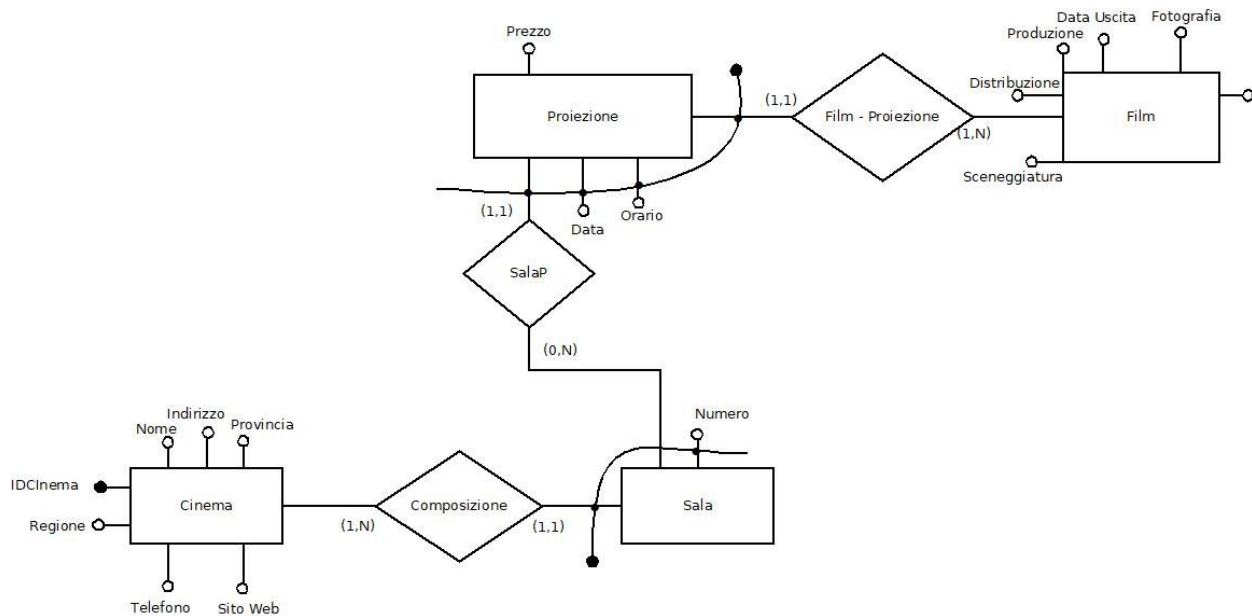
*RecitaF*(Film) referencia *Film*(CodiceFilm)

*RecitaF*(Attore) referencia *Attore*(IDAttore)

*RecitaSTV*(SerieTV) referencia *SerieTV*(CodiceSerie)

*RecitaSTV*(Attore) referencia *Attore*(IDAttore)





*Viene omessa una parte di schema ER relativa all'entità Contenuto perché già analizzata*

*Film*(CodiceFilm, Distribuzione, Musiche, Produzione, Data Uscita, Sceneggiatura, Fotografia, Durata)

*Cinema*(IDCinema, Nome, Indirizzo, Provincia, SitoWeb, Telefono, Regione)

*Sala*(Cinema, Numero)

*Proiezione*(Cinema, Orario, Sala, Film, Data, Prezzo)

*Sala*(Cinema) referencia *Cinema*(IDCinema)

*Proiezione*(Cinema, Sala) referencia *Sala*(Cinema, Numero)

## **MODELLO RELAZIONALE FINALE**

*Utente*(NomeUtente,Email, Password, ListaPreferiti\*, InizioCollaborazione\*, Tipologia)

*Contenuto*(IDContenuto, Nome, Paese, Genere, Anno )

*Voto*(Cardinalità, NumeroStelline, Contenuto\_Utente\*, EmailUtente\*)

*Preferenza*(EmailUtente, Utente, Contenuto)

*Regista*(IDRegista, Bio, Foto\*, Nome, Cognome, DataNascita, Età, LuogoNascita)

*Dirige*(Contenuto, Regista)

*Film*(CodiceFilm, Musiche, Distribuzione, Produzione, Data Uscita, Sceneggiatura, Fotografia, Durata)

*SerieTV*(CodiceSerie, NStagioni, NEpisodi, Produzione, Stato)

*ProgrammaTV*(CodiceProgramma, Descrizione, Stato)

*Episodio*(Stagione, Titolo, SerieTV, Durata, NumeroEpisodio, Sceneggiatore)

*DirigeEP*(Stagione, Titolo, SerieTV, Regista)

*Attore*(IDAttore, LuogoNascita, Foto\*, Nome, Cognome, DataNascita, Età, Bio)

*RecitaF*(Film, Attore, Personaggio)

*RecitaSTV*(SerieTV, Attore, Personaggio)

*Cinema*(IDCinema, Nome, Indirizzo, Provincia, SitoWeb, Telefono, Regione)

*Sala*(Cinema, Numero)

*Proiezione*(Cinema, Orario, Sala, Film, Data, Prezzo)

*CanaleProgr*(Nome, ProgrammaTV)

*PiattaformaProgr*(Nome, ProgrammaTV)

*CanaleSTV*(Nome, SerieTV)

*PiattaformaSTV*(Nome, SerieTV)

*RecentiRegista*(Regista, ContenutoRecente, Anno)

*FilmRecentiA*(Film, Attore, Anno)

*SerieTVRecentiA*(Serie, Attore, Anno)

### **Vincoli di integrità referenziale**

*Voto*(Utente,EmailUtente) referencia *Utente*(NomeUtente,Email)  
*Voto*(Contenuto) referencia *Contenuto*(IDContenuto)  
*Preferenza*(EmailUtente,Utente) referencia *Utente*(Email,NomeUtente)  
*Preferenza*(Contenuto) referencia *Contenuto*(IDContenuto)  
*Dirige*(Contenuto) referencia *Contenuto*(IDContenuto)  
*Dirige*(Regista) referencia *Regista*(IDRegista)  
*Film*(CodiceFilm) referencia *Contenuto*(IDContenuto)  
*SerieTV*(CodiceSerie) referencia *Contenuto*(IDContenuto)  
*ProgrammaTV*(CodiceProgramma) referencia *Contenuto*(IDContenuto)  
*Episodio*(SerieTV) referencia *SerieTV*(CodiceSerie)  
*DirigeEP*(Stagione,SerieTV,Titolo) referencia *Episodio*(Stagione,SerieTV,Titolo)  
*DirigeEP*(Regista) referencia *Regista*(IDRegista)  
*RecitaF*(Film) referencia *Film*(CodiceFilm)  
*RecitaF*(Attore) referencia *Attore*(IDAttore)  
*RecitaSTV*(SerieTV) referencia *SerieTV*(CodiceSerie)  
*RecitaSTV*(Attore) referencia *Attore*(IDAttore)  
*Sala*(Cinema) referencia *Cinema*(IDCinema)  
*Proiezione*(Cinema,Sala) referencia *Sala*(Cinema,Numero)  
*CanaleSTV*(SerieTV) referencia *SerieTV*(CodiceSerie)  
*CanaleProg*(ProgrammiTV) referencia *ProgrammaTV*(CodiceProgramma)  
*PiattaformaSTV*(SerieTV) referencia *SerieTV*(CodiceSerie)  
*PiattaformaProg*(ProgrammiTV) referencia *ProgrammaTV*(CodiceProgramma)

## Implementazione

### DDL creazione

```
BEGIN;  
drop table if exists FilmRecentiAttori;  
drop table if exists SerieTVRecentiAttori;  
drop table if exists contenutirecentiregisti;  
drop table if exists mediavoti;  
drop table if exists CanaleSTV;  
drop table if exists PiattaformaSTV;  
drop table if exists PiattaformaProgr;  
drop table if exists CanaleProgr;  
drop table if exists Proiezione;  
drop table if exists Sala;  
drop table if exists Cinema;  
drop table if exists RecitaSTV;  
drop table if exists RecitaF;  
drop table if exists Attore;  
drop table if exists DirigeEP;  
drop table if exists Episodio;  
drop table if exists ProgrammaTV;  
drop table if exists SerieTV;  
drop table if exists Film;  
drop table if exists Dirige;  
drop table if exists Regista;  
drop table if exists Preferenza;  
drop table if exists Voto;  
drop table if exists Contenuto;  
drop table if exists Utente;  
  
drop domain if exists TipologiaUtente;  
drop domain if exists Price;
```

```
drop domain if exists TipologiaUtente;  
drop domain if exists Price;  
drop domain if exists annocreazione;  
drop domain if exists stelline;  
drop domain if exists status;  
  
create domain TipologiaUtente  
as varchar(10) default 'Generico'  
check(value = 'Redattore' or value = 'Generico');
```

```
--Date formato YYYY-MM-DD
```

```
create table Utente (  
    NomeUtente varchar(20),  
    Email varchar(30),  
    Pass varchar(10) not null,  
    InizioCollaborazione date, --solo quando l'utente è redattore va aggiunta la data di inizio collaborazione  
    Tipologia TipologiaUtente,  
    check (case  
        when InizioCollaborazione IS NOT NULL  
        then 'Redattore' = Tipologia  
        else 'Generico' = Tipologia  
        end ),  
    primary key(NomeUtente,Email)  
);
```

```
--Supponiamo che non esistano contenuti più vecchi creati prima del 1800
```

```
create domain AnnoCreazione  
as numeric(4) default '2020'  
check(value >= 1800 );
```

```

--Date formato YYYY-MM-DD
create table Utente (
    NomeUtente varchar(20),
    Email varchar(30),
    Pass varchar(10) not null,
    InizioCollaborazione date, --solo quando l'utente è redattore va aggiunta la data di inizio collaborazione
    Tipologia TipologiaUtente,
    check (case
        when InizioCollaborazione IS NOT NULL
        then 'Redattore' = Tipologia
        else 'Generico' = Tipologia
        end ),
    primary key(NomeUtente,Email)
);

--Supponiamo che non esistano contenuti più vecchi creati prima del 1800
create domain AnnoCreazione
as numeric(4) default '2020'
check(value >= 1800 );

--IDContenuto S.. per le serie TV, P.. per i programmi e F.. per i film
create table Contenuto (
    IDContenuto varchar(20) unique,
    Nome varchar(50) not null unique,
    Paese varchar(20) not null,
    Genere varchar(50) not null,
    Anno AnnoCreazione not null,
    unique(IDContenuto,Nome),
    primary key(IDContenuto)
);

```

```

create domain Stelline
as integer default NULL
check (value>=1 and value<=5);

create table Voto(
    Cardinalità SERIAL primary key,
    NumeroStelline Stelline,
    Contenuto varchar(20), --Referenzia IDContenuto di Contenuto
    Utente varchar(20), --può essere null
    EmailUtente varchar(20), --può essere null
    unique(Utente,EmailUtente,Contenuto), --ogni utente può esprimere un solo voto per ogni contenuto
    foreign key(Utente,EmailUtente) references Utente(NomeUtente,Email)
        on update cascade
        on delete set null,
    foreign key (Contenuto) references Contenuto(IDContenuto)
        on update cascade
        on delete cascade
);

create table Preferenza(
    EmailUtente varchar(20),
    Utente varchar(20),
    Contenuto varchar(20),
    primary key(EmailUtente, Utente, Contenuto),
    foreign key (EmailUtente,Utente) references Utente(Email,NomeUtente)
        on update cascade
        on delete cascade,
    foreign key (Contenuto) references Contenuto(IDContenuto)
        on update cascade
        on delete cascade --Cancello a cascata
);

```



```

create table Regista(
  IDRegista SERIAL primary key,
  Bio varchar(2000) not null,
  Foto bytea, --bytea è il tipo per le immagini, può essere null
  Nome varchar(30) not null,
  Cognome varchar(30) not null,
  DataNascita date not null, --formato YYYY-MM-DD
  Età integer not null,
  LuogoNascita varchar(30) not null,
  unique(Nome,Cognome,DataNascita,LuogoNascita)
);

create table Dirige(
  Contenuto varchar(20),
  Regista integer,
  primary key (Contenuto,Regista),
  foreign key (Regista) references Regista(IDRegista)
    on update cascade
    on delete cascade, --se elimino un regista allora elimino anche i contenuti che ha diretto
  foreign key (Contenuto) references Contenuto(IDContenuto)
    on update cascade
    on delete cascade --se elimino un contenuto allora elimino anche la sua associazione con un regista
);

```

```

create table Film(
  CodiceFilm varchar(20) primary key, --referenzia Contenuto
  DataUscita date not null,
  Distribuzione varchar(20) not null,
  Sceneggiatura varchar(20) not null,
  Fotografia varchar(20) not null,
  Durata integer not null,
  Musiche varchar(20) not null,
  Produzione varchar(50) not null,

  foreign key (CodiceFilm) references Contenuto(IDContenuto)
    on update cascade
    on delete cascade --se elimino il film nei contenuti allora viene eliminato anche qua
);

create domain Status
as varchar(15) default 'In Produzione'
check (value like 'In Produzione' or value like 'Conclusa');

create table SerieTV(
  CodiceSerie varchar(20), --referenzia contenuto
  NStagioni integer not null,
  NEpisodi integer not null,
  Produzione varchar(50) not null,
  Stato Status not null,

  primary key(CodiceSerie),
  foreign key (CodiceSerie) references Contenuto(IDContenuto)
    on update cascade
    on delete cascade
);

```

```

create table ProgrammaTV(
    CodiceProgramma varchar(20) primary key, --referenzia contenuto
    Descrizione varchar(2000) not null,
    Stato Status,
    foreign key (CodiceProgramma) references Contenuto(IDContenuto)
        on update cascade
        on delete cascade --programma viene eliminato a cascata
);

--Il numero di episodio e numero di stagione dev'essere < di NEpisodio e NStagioni di SerieTV
--per verificare questo si può ricorrere ad un vincolo con sottointerrogazione ma in PostgreSQL non sono supportati.

create table Episodio(
    SerieTV varchar(20), --referenzia Contenuto
    Stagione integer,
    Titolo varchar(40),
    Durata time not null,
    NumeroEpisodio integer not null,
    Sceneggiatore varchar(30) not null,
    primary key(SerieTV,Titolo,Stagione),
    foreign key (SerieTV) references SerieTV(CodiceSerie)
        on update cascade
        on delete cascade --se una serie viene eliminata, i suoi contenuti sono eliminati a cascata
);

create table RecitaF(
    Film varchar(20),
    Attore integer,
    Personaggio varchar(20) not null,
    primary key(Film,Attore),
    foreign key (Film) references Film(CodiceFilm)
        on update cascade
        on delete cascade,
    foreign key (Attore) references Attore(IDAttore)
        on update cascade
        on delete cascade
);

create table RecitaSTV(
    SerieTV varchar(20),
    Attore integer,
    Personaggio varchar(20) not null,
    primary key(SerieTV,Attore),
    foreign key (Attore) references Attore(IDAttore)
        on update cascade
        on delete cascade, --Quando viene eliminato l'attore dal database spariscono tutti i contenuti a cui ha preso parte
    foreign key (SerieTV) references SerieTV(CodiceSerie)
        on update cascade
        on delete cascade --Quando viene eliminato l'attore dal database spariscono tutti i contenuti a cui ha preso parte
);

```

```

create table Cinema(
    IDCinema SERIAL primary key,
    Nome varchar(30) not null,
    Indirizzo varchar(50) not null,
    Provincia varchar(20) not null,
    Telefono integer not null,
    SitoWeb varchar(50), -- ho aggiunto i contatti a cinema
    Regione varchar(20) not null,
    unique(Nome,Indirizzo,Provincia)
);

create table Sala(
    Cinema integer,
    Numero integer,
    primary key(Cinema, Numero),

    foreign key (Cinema) references Cinema(IDCinema)
        on update cascade
        on delete cascade --quando elimino un cinema elimino anche le sue sale
);

```

```

create domain Price
as decimal(5,2)
check (value >=0);

```

```

create table Proiezione(
    Cinema integer,
    DataP date,
    Orario time,
    Film varchar(20) not null,
    Prezzo Price not null,
    Sala integer not null,
    primary key(Cinema,Sala,Film,Orario,DataP),
    foreign key (Cinema,Sala) references Sala(Cinema,Numero)
        on update cascade
        on delete cascade
);

create table CanaleProgr(
    Nome varchar(20),
    ProgrammaTV varchar(2),
    primary key(Nome,ProgrammaTV),
    foreign key (ProgrammaTV) references ProgrammaTV(CodiceProgramma)
        on update cascade
        on delete cascade
);

```



```

create table PiattaformaProgr(
    Nome varchar(20),
    ProgrammaTV varchar(20),
    primary key(Nome,ProgrammaTV),
    foreign key (ProgrammaTV) references ProgrammaTV(CodiceProgramma)
        on update cascade
        on delete no action
);

create table CanaleSTV(
    Nome varchar(20),
    SerieTV varchar(20),
    primary key(Nome,SerieTV),
    foreign key (SerieTV) references SerieTV(CodiceSerie)
        on update cascade
        on delete cascade
);

create table PiattaformaSTV(
    Nome varchar(20),
    SerieTV varchar(20),
    primary key(Nome,SerieTV),
    foreign key ( SerieTV) references SerieTV(CodiceSerie)
        on update cascade
        on delete cascade
);

```

## DML di popolamento

```

insert into Utente(NomeUtente,Email,Pass) values ('Pippo01','Pippo01@hotmail.com','12nc');
insert into Utente(NomeUtente,Email,Pass,Tipologia,InizioCollaborazione) values ('Maria','Mariag@hotmail.com','23219oo0','Redattore','2012-01-12');

insert into Contenuto(IDContenuto,Nome,Paese,Genere,Anno)
values ('F1','Star Wars Episodio III: La vendetta dei Sith','Usa','Fantascienza',2005);

insert into Contenuto(IDContenuto,Nome,Paese,Genere,Anno)
values ('F2','Titanic','Usa','Drammatico',1997);

insert into Contenuto(IDContenuto,Nome,Paese,Genere,Anno)
values('S1','La casa di carta','Spagna','drammatico',2017);

insert into Contenuto(IDContenuto,Nome,Paese,Genere,Anno)
values('S2','The Office','UK','commedia',2015); --data modificata per provare i contenuti recenti dei registi

insert into Contenuto(IDContenuto,Nome,Paese,Genere,Anno)
values ('S3','Dr House','USA','medico',2004);

Insert into Contenuto(IDContenuto,Nome,Paese,Genere,Anno)
values ('P1','Zelig','Italia','varietà',1997);

Insert into Contenuto(IDContenuto,Nome,Paese,Genere,Anno)
values ('F3','Star Wars Episodio IX:L ascesa di Skywalker','USA','Fantascienza',2019);

Insert into Contenuto(IDContenuto,Nome,Paese,Genere,Anno)
values ('P2','Camera caffè','Italia','sitcom',2003);

```

```

insert into Film(CodiceFilm,DataUscita,Musiche,Produzione,Durata,Distribuzione,Sceneggiatura,Fotografia)
values ('F1','2005-05-20','John Williams','Lucas Film',140,'20th Century Fox','George Lucas','David Tattersall');

insert into Film(CodiceFilm,DataUscita,Musiche,Produzione,Durata,Distribuzione,Sceneggiatura,Fotografia)
values ('F2','1997-11-1','James Horner','Paramount Pictures',195,'20th Century Fox','James Cameron','Russell Carpenter');

insert into Film(CodiceFilm,DataUscita,Musiche,Produzione,Durata,Distribuzione,Sceneggiatura,Fotografia)
values ('F3','2019-12-14','John Williams','Disney',142,'Disney','J.J Abrams','Daniel Midell');

insert into Serietv(CodiceSerie,NStagioni,NEpisodi,Produzione)
values ('S1','3','31','Atresmedia, Netflix');

insert into Serietv(CodiceSerie,NStagioni,NEpisodi,Produzione,Stato)
values ('S2','9','120','NBC Universal Television','Conclusa');

insert into Serietv(CodiceSerie,NStagioni,NEpisodi,Produzione,Stato)
values ('S3','8','177','Universal Media Studios (UMS)','Conclusa');

insert into programmatv (CodiceProgramma,Descrizione,Stato)
values ('P1','CABARET COMICO','Conclusa');
insert into programmatv (CodiceProgramma,Descrizione,Stato)
values ('P2','SITCOM','Conclusa');

insert into Regista(Bio,Foto,Nome,Cognome,DataNascita,Età,LuogoNascita)
values ('è un regista, sceneggiatore, produttore cinematografico e montatore statunitense.
È stato proprietario di una holding del cinema, la Lucasfilm. È il creatore delle saghe cinematografiche di Guerre stellari e Indiana Jones.',
-bytea('/home/letizia/Downloads/george-lucas-biography_2.jpeg'),'George','Lucas','1944-05-14',76,'USA'); --regista di star wars

```

```

insert into Regista(Bio,Foto,Nome,Cognome,DataNascita,Età,LuogoNascita)
values ('è un regista, sceneggiatore, produttore cinematografico e montatore canadese. I suoi due ultimi film,
Avatar e Titanic, sono rispettivamente il secondo ed il terzo film di maggiore incasso della storia del cinema.
I suoi lavori da regista hanno incassato complessivamente 5,2 miliardi di dollari.',
-bytea('/home/letizia/Downloads/9682514.jpeg'),'James','Cameron','1954-08-16',65,'Canada'); --regista Titanic

insert into Regista(Bio,Foto,Nome,Cognome,DataNascita,Età,LuogoNascita)
values('','','Kenneth William','Kwapis','1957-08-17',62,'East St.Louis');
insert into Regista(Bio,Foto,Nome,Cognome,DataNascita,Età,LuogoNascita)
values('','','Bryan Jay','Singer','1965-09-17',54,'New York City');

insert into Regista(Bio,Foto,Nome,Cognome,DataNascita,Età,LuogoNascita)
values ('è un regista, sceneggiatore, produttore cinematografico statunitense',
-bytea('/home/letizia/Downloads/9682514.jpeg'),'Jeffrey Jacob','Adams','1966-08-16',53,'USA'); --regista Tsw episodio 9

insert into Attore(LuogoNascita,Nome,Cognome,DataNascita,Età,Bio)
values('Los Angeles','Leonardo','DiCaprio','1974-11-11','45','attore protagonista titanic');
insert into Attore(LuogoNascita,Nome,Cognome,DataNascita,Età,Bio)
values ('Madrid','María','Pedraza','1996-01-26','24','attrice e ballerina');
insert into Attore(LuogoNascita,Nome,Cognome,DataNascita,Età,Bio)
values ('San Diego','Adam','Driver','1983-11-19',36,'USA');

start transaction;
insert into Episodio ( SerieTV, Stagione,Titolo,Durata,NumeroEpisodio,Sceneggiatore)
values ('S2',1,'Un Capo Esempiare','00:23:00',1,'Ricky Gervais');
insert into Episodio ( SerieTV, Stagione,Titolo,Durata,NumeroEpisodio,Sceneggiatore)
values ('S2',1,'Integrazione','00:21:00',2,'B.J.Novak');
insert into Episodio ( SerieTV, Stagione,Titolo,Durata,NumeroEpisodio,Sceneggiatore)
values ('S3',1,'Una prova per non morire','00:30:00',1,'David Shore');

```



```

insert into Voto(NumeroStelline,Contenuto,Utente,EmailUtente) values (5,'S1','Maria','Mariag@hotmail.com');
insert into Voto(NumeroStelline,Contenuto,Utente,EmailUtente) values (4,'F1','Pippo01','Pippo01@hotmail.com');
insert into Voto(NumeroStelline,Contenuto,Utente,EmailUtente) values (2,'F2','Pippo01','Pippo01@hotmail.com');
insert into Voto(NumeroStelline,Contenuto,Utente,EmailUtente) values (5,'P1','Pippo01','Pippo01@hotmail.com');
insert into Voto(NumeroStelline,Contenuto,Utente,EmailUtente) values (3,'F1','Maria','Mariag@hotmail.com');

insert into Preferenza(EmailUtente,Utente,Contenuto) values ('Mariag@hotmail.com','Maria','F1');
insert into Preferenza(EmailUtente,Utente,Contenuto) values ('Pippo01@hotmail.com','Pippo01','S1');

insert into Dirige (Contenuto,registra)
values ('F1', 1);
insert into Dirige (Contenuto,registra)
values ('F2', 2);
insert into Dirige (Contenuto,registra)
values ('F3', 5);
insert into Dirige (Contenuto,registra)
values ('S3',4);

insert into DirigeEp (Stagione,Titolo,SerieTV,Regista )
values (1,'Un Capo Esempolare','S2',3);
insert into DirigeEp (Stagione,Titolo,SerieTV,Regista )
values (1,'Integrazione','S2',3);
insert into DirigeEp (Stagione,Titolo,SerieTV,Regista )
values (1,'Una prova per non morire','S3',4);

```

```

insert into RecitaF (Film,Attore,Personaggio)
values ('F2',1,'Jack Dawson');
insert into RecitaF(Film,Attore,Personaggio)
values('F3',3,'Kylo Ren/Ben Solo');

insert into Recitastv(Serietv,Attore,Personaggio)
values ('S1',2,'Alison Parker');

insert into Cinema(Nome,Indirizzo,Provincia,Telefono,SitoWeb,Regione)
values ('UCI Cinemas Torino Lingotto','Via Nizza 262','Torino','011892960','','Piemonte' );
insert into Cinema(Nome,Indirizzo,Provincia,Telefono,SitoWeb,Regione)
values ('Cinema Odeon Firenze',' Piazza degli Strozzi','Firenze','055214068','','Toscana' );
insert into Sala(Cinema,Numero)
values (1,5);
insert into Sala(Cinema,Numero)
values (2,8);

insert into Proiezione (Cinema,DataP,Orario,Film,Prezzo,Sala)
values(1,'2020-08-20','14:20','F2',5.10,5);
insert into Proiezione (Cinema,DataP,Orario,Film,Prezzo,Sala)
values(2,'2005-05-20','15:00','F1',6.10,8);

insert into PiattaformaSTV (Nome,SerieTV)
values ('Netfilx','S1');
insert into PiattaformaSTV (Nome,SerieTV)
values ('Prime Video','S2');
insert into PiattaformaSTV (Nome,SerieTV)
values ('Prime Video','S3');

```

```

insert into CanaleProgr(Nome,ProgrammaTV )
    values ('Canale5','P1');
insert into CanaleProgr(Nome,ProgrammaTV )
    values ('Italia1','P2');

insert into CanaleSTV(Nome,SerieTV )
    values ('Sky','S3');

create table ContenutiRecentiRegisti as
select r.nome, r.cognome, c.idcontenuto ,c.nome ContenutoEpisodio
from dirige d join contenuto c on (d.Contenuto = c.idcontenuto)
    join regista r on (d.regista = r.idregista)
where DATE_PART('year',current_date) - c.Anno <=10

union --aggiungo anche gli episodi diretti

select r.nome, r.cognome, c.idcontenuto, ep.Titolo ContenutoEpisodio
from dirigeep ep join serietv stv on (ep.serietv = stv.codiceserie)
    join regista r on (ep.regista = r.idregista)
    join contenuto c on (c.idcontenuto = stv.codiceserie)
where DATE_PART('year',current_date) - c.Anno <=10;

create table SerieTvRecentiAttori as
select a.Nome,a.Cognome, c.Nome SerieTV, rstv.Personaggio
from recitastv rstv join contenuto c on (rstv.SerieTV = c.idcontenuto)
    join attore a on (rstv.attore = a.idattore)
where DATE_PART('year',current_date) - c.Anno <=10;

```

```

create table SerieTvRecentiAttori as
select a.Nome,a.Cognome, c.Nome SerieTV, rstv.Personaggio
from recitastv rstv join contenuto c on (rstv.SerieTV = c.idcontenuto)
    join attore a on (rstv.attore = a.idattore)
where DATE_PART('year',current_date) - c.Anno <=10;

create table FilmRecentiAttori as
select a.Nome, a.Cognome ,c.Nome Film, rf.Personaggio PersonaggioInterpretato
from recitaf rf join contenuto c on (rf.film = c.idcontenuto)
    join attore a on (rf.attore = a.idattore)
where DATE_PART('year',current_date) - c.Anno <=10;

create table MediaVoti as
select c.IDContenuto Codice, c.Nome Titolo, avg(umerostelline)
from Contenuto c join voto v on (c.IDContenuto = v.Contenuto)
group by c.IDContenuto, c.Nome;

```

**Commento:** si considera recente un contenuto che è stato pubblicato meno di 10 anni fa, se un attore/regista ha contenuti più vecchi di 10 anni allora non viene mostrato nella tabella.

## Operazioni

```
--Test
insert into Voto(NumeroStelline,Contenuto,Utente,EmailUtente) values (1,'F100','Pippo01','Pippo01@hotmail.com');--da errore perchè non esiste un film con
insert into Voto(NumeroStelline,Contenuto,Utente,EmailUtente) values (4,'F2','Mela','Mela00@hotmail.com'); --da errore perchè non esiste un utente Mela
insert into Voto(NumeroStelline,Contenuto,Utente,EmailUtente) values (5,'S2','Pippo01','Pippo01@hotmail.com'); --da errore perchè non esiste un email così
insert into Utente(NomeUtente,Email,Pass) values ('Pippo01','Pippo01@hotmail.com','12cccc'); --da errore perchè c'è già un utente con quel nome
insert into Utente(NomeUtente,Email,Pass) values ('Pippo','Pippo01@hotmail.com','12nc'); --da errore perchè esiste già un email così

--Test utente generico sulla data di inizio collaborazione
insert into Utente(NomeUtente,Email,Pass,Tipologia,InizioCollaborazione) values ('Claudia','Claudia11@hotmail.com','2gskw!','Generico','2012-01-12');

--Per prova preferenze e voti:
delete from Utente where NomeUtente = 'Maria';
delete from Utente where NomeUtente = 'Pippo01';

--Update utente
update utente
set Email = 'Pippo@gmail.com'
where Email = 'Pippo01@hotmail.com';*/

--Update contenuto
update contenuto
set idcontenuto = 'S10'
where idcontenuto = 'S1';

--Eliminazione di un contenuto
delete from Contenuto where idcontenuto = 'F1';

--Eliminazione regista
delete from Regista where idregista = 1;
```

```
--Per prova preferenze e voti:
delete from Utente where NomeUtente = 'Maria';
delete from Utente where NomeUtente = 'Pippo01';

--Update utente
update utente
set Email = 'Pippo@gmail.com'
where Email = 'Pippo01@hotmail.com';*/

--Update contenuto
update contenuto
set idcontenuto = 'S10'
where idcontenuto = 'S1';

--Eliminazione di un contenuto
delete from Contenuto where idcontenuto = 'F1';

--Eliminazione regista
delete from Regista where idregista = 1;

--Aggiornamento regista
update Regista
set idregista = 10
where idregista = 4;

--Eliminazione Cinema
delete from Cinema where idcinema = 1;

commit;
```