

# Progetto Database

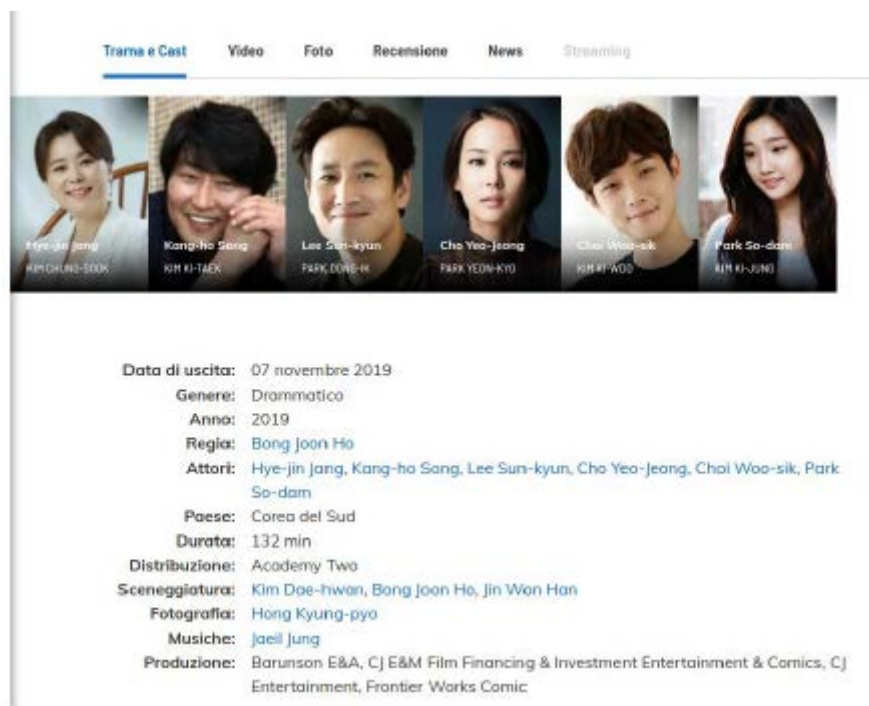
Samuel Girardello (mat. 886317)

Marco Demichelis (mat. 883814)

## 1. Progettazione Concettuale

### 1.1. Requisiti iniziali

Si vuole realizzare una base di dati per la gestione di una piattaforma che fornisce informazioni su film in uscita nei cinema e su serie e programmi TV in onda, liberamente ispirato a piattaforme come ComingSoon. Un numero limitato di utenti della redazione si occupa di aggiornare le informazioni sui contenuti disponibili inserendo i dati di film e serie, comprese le date di uscita e programmazione in sala e in TV. Degli utenti si memorizzano nome utente, password e indirizzo email. Per i redattori si tiene traccia anche della data di inizio collaborazione.



I contenuti hanno una serie di caratteristiche come evidenziato in Figura 1

I contenuti hanno una serie di caratteristiche come evidenziato in Figura 1. In particolare, a ogni film e serie TV è associata una lista di attori, con l'indicazione del personaggio interpretato. Notare che alcune informazioni tipiche dei film, ad esempio la regia, per le serie TV sono specifiche dei singoli episodi. Attori e registi hanno una propria scheda (esempio in Figura 2) dove vengono visualizzati nome, foto, e alcune informazioni anagrafiche. Qui può essere anche mostrata una lista dei contenuti più recenti a cui la persona ha partecipato.



## 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),

Figura 2: Dati di un attore/regista

I film vengono proiettati in svariati cinema in tutto il paese. Per ogni cinema, si tiene traccia del suo nome, contatti, e localizzazione (regione, provincia e indirizzo). La proiezione avviene a degli orari specifici in diverse date, in una delle sale disponibili al cinema, e comporta il pagamento di un biglietto il cui prezzo è indicato sul sito.

Programmi e serie TV vanno invece 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, es. commedia, drammatico, etc. In più, i serial 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 testuale.

#### Stagione 1<sup>5</sup>

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

Figura 3: Stagioni ed episodi

I visitatori del sito, previa registrazione tramite email e password, possono lasciare un voto (da 1 a 5 stelline) su qualunque contenuto. La media dei voti ricevuti da film e programmi viene mostrata nelle rispettive schede. Inoltre, gli utenti possono cliccare su un link “inserisci tra i preferiti” presente nelle schede dei contenuti, in modo da poter visualizzare in seguito i contenuti salvati. Nel momento della cancellazione di un utente dal sito, i suoi preferiti possono essere rimossi dal database, mentre i voti espressi dal visitatore rimangono in memoria.

## 1.2. Glossario dei termini

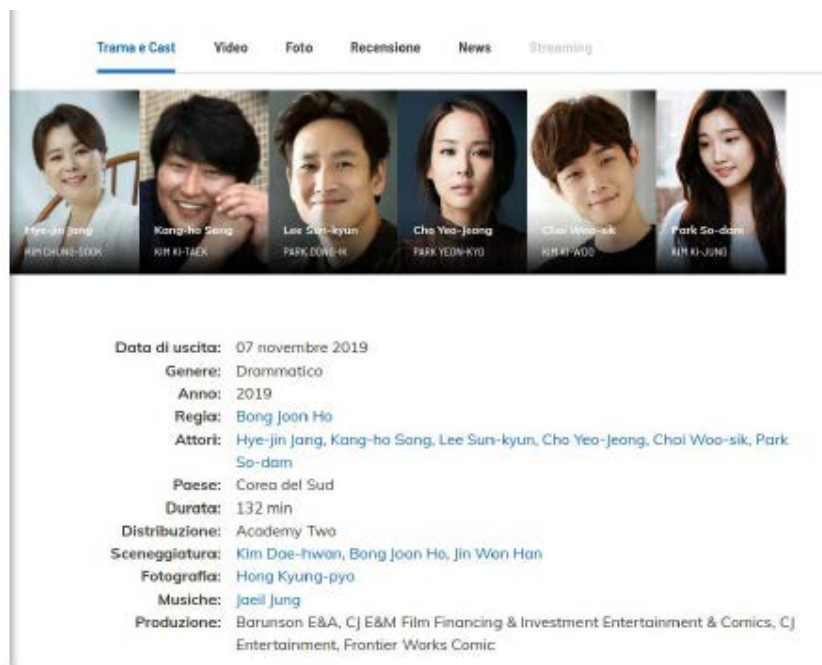
Termine	Descrizione	Sinonimi	Collegamenti
Utenti	Gli utenti finali registrati al sito.	Visitatori	Contenuti, voto, preferiti
Redattori	Gli utenti gestori del sito.	Utenti della redazione	Contenuti
Contenuti	Contenuti rappresentati nel sito. Possono essere film o Serie tv o programmi.		Attori e registi, cinema, piattaforme, voto, preferiti
Film	Rappresentano i contenuti di tipo Film.		Attori e registi, cinema, voto, preferiti
Serie TV	Rappresentano i contenuti di tipo serie TV	Serial	Attori e registi, piattaforme, voto, preferiti
Programmi	Rappresentano i contenuti di tipo Programmi		Piattaforme, voto, preferiti
Attori e registi	Rappresentano i personaggi del cinema che hanno partecipato ai contenuti svolgendo il ruolo di attori, registi e sceneggiatori.		Contenuti
Cinema	Sono i cinema dove vengono proiettati i film		Contenuti, proiezione, sala
Sala	Rappresentano le sale di un Cinema. Possono essere disponibili o non disponibili.		Cinema, proiezione
Proiezione	Rappresentano le proiezioni al cinema dei film.		Contenuti, cinema, sala
Piattaforme	Sono le piattaforme dove vengono trasmessi i programmi tv e le serie. Possono essere canali TV o piattaforme di streaming		Contenuti
Episodi	Rappresentano gli episodi di una serie TV, appartenenti a una		Stagione

	specifica stagione		
Stagioni	Rappresentano le stagioni in cui una serie TV è racchiusa		Serie TV, Episodi
Voti	Rappresentano i voti rilasciati dagli utenti utilizzatori della piattaforma.		Utente, contenuti
Preferiti	Rappresentano i contenuti selezionati come preferiti da parte dell'utente utilizzatore della piattaforma	Contenuti salvati	Contenuti, utente

### 1.3. Requisiti rivisti e strutturati in gruppi di frasi omogenee

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

Un numero limitato di utenti della redazione **1 redattori** si occupano di aggiornare le informazioni sui contenuti disponibili inserendo **inserire** i dati di film e serie, comprese le date di uscita e programmazione in sala e in TV. Degli utenti si memorizzano **rappresentiamo** nome utente, password e indirizzo email. Per i redattori si tiene traccia anche della **rappresentiamo anche** data di inizio collaborazione.



I contenuti hanno una serie di caratteristiche come evidenziato in Figura 1

In particolare, a **Per** ogni film e serie TV è associata **rappresentiamo** una lista di attori, con l'indicazione del personaggio interpretato. ~~Notare che alcune informazioni tipiche dei film, ad esempio la regia, per le serie TV sono specifiche dei singoli episodi~~ **Per le serie TV, per ogni singolo episodio rappresentiamo i registi.** Di **Degli** attori e registi **artisti** hanno una propria scheda (esempio in Figura 2) dove vengono visualizzati

rappresentiamo nome, foto, e alcune informazioni anagrafiche età, luogo di nascita, data di nascita e anche. Qui può essere anche mostrata una lista dei contenuti, i contenuti più recenti a cui la persona ha partecipato.



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),

Figura 2: Dati di un attore/regista

I film vengono proiettati in svariati molti cinema in tutto il paese al cinema. Per ogni cinema, si tiene traccia del suo rappresentiamo nome, contatti, e localizzazione luogo (regione, provincia e indirizzo). La Per ogni proiezione avviene a degli orari specifici in diverse date rappresentiamo la data, l'ora, in una delle sale disponibili al cinema la sala del cinema (ove questa sia disponibile), e comporta il pagamento di un biglietto il cui prezzo è indicato sul sito e il prezzo del biglietto.

Programmi e serie TV vanno invece in onda sono trasmessi su diverse piattaforme (es. Netflix, Rai Gulp,...), che possono essere canali TV tradizionali o piattaforme di streaming video. Le Delle serie TV e dei film sono organizzate rappresentiamo anche il per genere, come per i film, es. commedia, drammatico, etc. In più, i serial Le serie TV sono composti composte da episodi, racchiusi in una o più stagioni, dove. Per ogni episodio ha uno specifico rappresentiamo titolo, durata, regista, e sceneggiatore. Anche per Per i programmi sono memorizzate alcune informazioni, come rappresentiamo l'anno di messa in onda, genere, paese, durata e una descrizione testuale.

#### Stagione 1

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

Figura 3: Stagioni ed episodi



I visitatori del sito, previa registrazione tramite email e password **gli utenti**, possono lasciare un voto (da 1 a 5 stelline) su qualunque contenuto. La media dei voti ricevuti da film e programmi viene mostrata nelle rispettive schede. **Per ogni contenuto rappresentiamo anche la media dei voti**. Inoltre, gli utenti possono cliccare su un link “inserisci tra i preferiti” presente nelle schede dei contenuti, in modo da poter visualizzare **il contenuto** in seguito **nei contenuti salvati preferiti**. Nel momento della cancellazione di un utente dal sito, i suoi preferiti possono essere rimossi dal database, mentre i voti espressi dal visitatore **dagli utenti** rimangono in memoria.

### 1.3.1. Gruppi di frasi omogenee

#### FRASI DI CARATTERE GENERALE

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

#### FRASI RELATIVE A TIPI SPECIFICI DI UTENTI

I redattori si occupano di inserire i dati di film e serie, comprese le date di uscita e programmazione in sale e TV. Degli utenti rappresentiamo nome utente, password e indirizzo email. Per i redattori rappresentiamo anche data di inizio collaborazione. Gli utenti, 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 nelle schede dei contenuti , in modo da poter visualizzare il contenuto, in seguito nei preferiti. Nel momento della cancellazione di un utente dal sito, i suoi preferiti possono essere rimossi dal database, mentre i voti espressi dagli utenti rimangono in memoria.

#### FRASI RELATIVE A TIPI SPECIFICI DI CONTENUTI

Per ogni film e serie TV rappresentiamo una lista di attori con l’indicazione del personaggio interpretato. Per le serie TV, per ogni singolo episodio rappresentiamo i registi. I film vengono proiettati al cinema. Programmi e serie TV sono trasmessi su diverse piattaforme (es. Netflix, Rai Gulp,...), che possono essere canali TV o piattaforme di streaming. Delle serie TV e dei film rappresentiamo anche il genere (es commedia, drammatico, etc..). Le serie TV sono composte da episodi, racchiusi in una o più stagioni. Per ogni episodio rappresentiamo titolo, durata, regista, e sceneggiatore. Per i programmi rappresentiamo l’anno di messa in onda, genere, paese, durata e una descrizione testuale. Per ogni contenuto rappresentiamo anche la media dei voti.

#### FRASI RELATIVE AGLI ARTISTI

Degli artisti rappresentiamo nome, foto, età, luogo di nascita, data di nascita e anche i contenuti più recenti a cui la persona ha partecipato.

#### FRASI RELATIVE ALLE PIATTAFORME

Possono essere canali TV o piattaforme streaming.

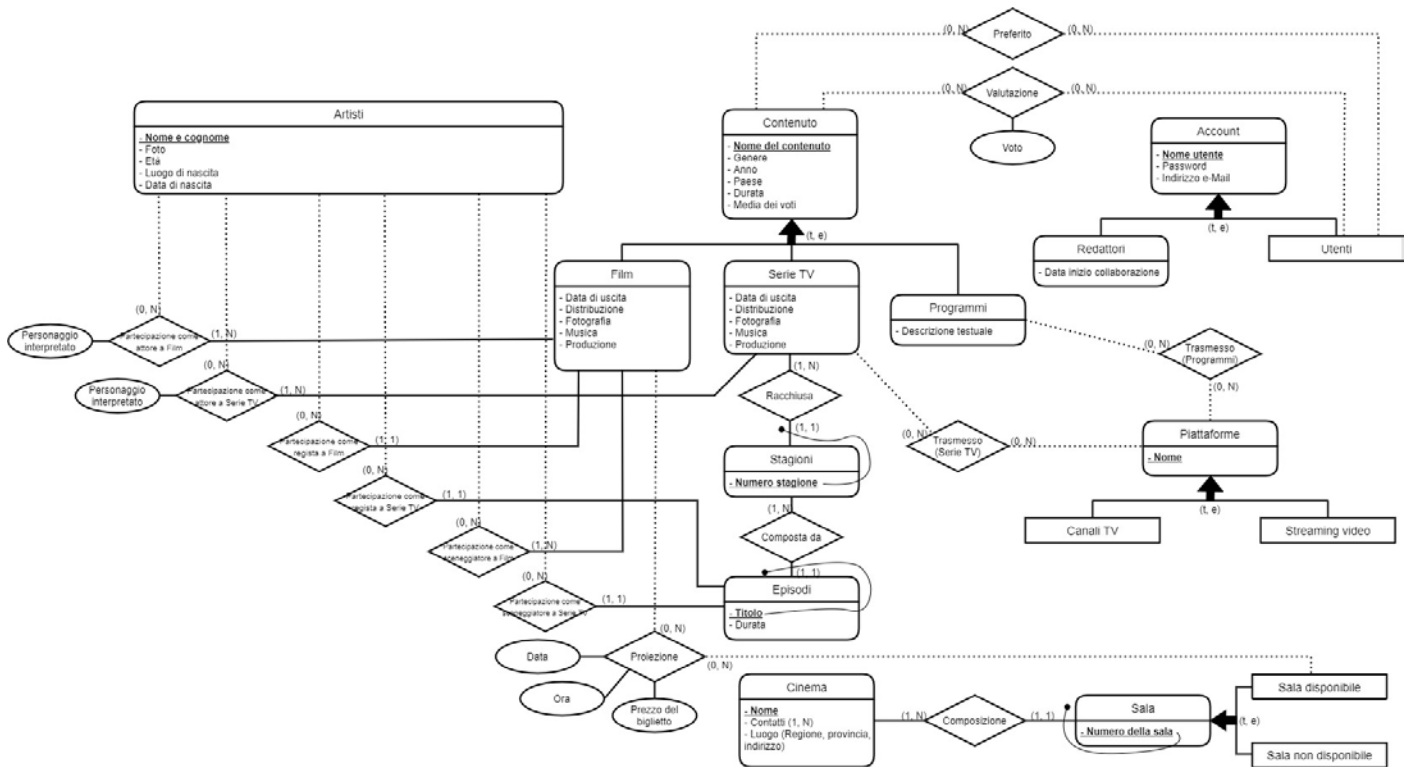
#### FRASI RELATIVE ALLE PROIEZIONI

Per ogni proiezione rappresentiamo la data, l’ora, la sala del cinema (ove questa sia disponibile) e il prezzo del biglietto.

#### FRASI RELATIVE AL CINEMA

Per ogni cinema, rappresentiamo nome, contatti, luogo (regione, provincia e indirizzo).

## 1.4. Schema E-R + business rules



### Business rule

- 1) I contenuti recenti associati ad un artista sono un sottoinsieme dei contenuti a cui quest'ultimo ha partecipato, i quali contenuti non hanno una data di uscita non più vecchia di un'anno.
- 2) Un artista può partecipare allo stesso contenuto come attore, oppure come regista, oppure come sceneggiatore.
- 3) La media dei voti di un contenuto si riferisce alla media delle valutazioni di quel contenuto all'interno dell'associazione "Valutazione".

### 1.4.1. Business rules

- 1) I contenuti recenti associati ad un artista sono un sottoinsieme dei contenuti a cui quest'ultimo ha partecipato, i quali contenuti non hanno una data di uscita non più vecchia di un'anno.
- 2) Un artista può partecipare allo stesso contenuto come attore, oppure come regista, oppure come sceneggiatore.
- 3) La media dei voti di un contenuto si riferisce alla media delle valutazioni di quel contenuto all'interno dell'associazione "Valutazione".

## 2. Progettazione logica

### 2.1. Tavola dei volumi

CONCETTO	TIPO	VOLUME
Artisti	E	4000 (ipotetica quantità di pers. del cinema)
Contenuto	E	50.000 (ipotetica quantità di contenuti)
Film	E	48.900 (97,8% dei contenuti)
Serie TV	E	1000 (2% dei contenuti)
Programmi	E	100 (0,2% dei contenuti)
Stagioni	E	2000 (in media 2 stagioni per serie TV)
Episodio	E	20.000 (in media 10 episodi a stagione)
Account	E	500.000 (ipotetico numero di account)
Redattori	E	100 (0,02% di redattori)
Utenti	E	499.900 (99,97% di utenti visitatori)
Piattaforme	E	100
Canali TV	E	30 (30% delle piattaforme)
Streaming video	E	70 (70% delle piattaforme)
Cinema	E	50
Sala	E	250 (in media 5 sale a cinema)
Sala disponibile	E	100 (in media 2 sale disponibili a cinema)
Sala non disponibile	E	150 (in media 3 sale non disponibili a cinema)
Preferito	A	2.500.000 (circa 5 preferiti per utente)
Valutazione	A	10.000.000 (circa 21 valutazioni per utente)
Partecipazione come attore a film	A	500.000 (circa 10 attori per film)
Partecipazione come attore a serie TV	A	30.000 (circa 30 attori per serie TV)
Partecipazione come regista a film	A	48.900 (un regista per film)
Partecipazione come regista a serie TV	A	20.000 (un regista per episodio)
Partecipazione come sceneggiatore a film	A	244.500 (circa 5 sceneggiatori a film)
Partecipazione come sceneggiatore a serie TV	A	20.000 (uno sceneggiatore per episodio)
Racchiusa	A	2000 (n. di stagioni)
Composta da	A	20.000 (n. di episodi)



Trasmesso (serie TV)	A	2.000 (alcune serie TV non sono trasmesse, altre sono trasmesse su più piattaforme)
Trasmesso (programmi)	A	200 (alcuni programmi non sono trasmessi, altri sono trasmessi su più piattaforme)
Proiezione	A	3.500 (2 proiezioni per sala al giorno, 500 proiezioni al giorno, dopo la proiezione l'informazione viene cancellata, mediamente rimane in memoria per una settimana)
Composizione	A	250 (n. delle sale)

(Un artista può partecipare allo stesso contenuto come attore, oppure come regista, oppure come sceneggiatore)

## 2.2. Tavola delle operazioni

Operazione	Descrizione	Tipo	Frequenza
1	Consulta dati di un contenuto da parte di un utente	I	50.000 volte al giorno
2	Consulta dati di un personaggio del cinema (Artisti) da parte di un utente	I	10.000 volte al giorno
3	Inserimento dati di un contenuto da parte dei redattori	I	200 volte al giorno
4	Inserimento dati di un personaggio del cinema (Artisti) da parte dei redattori	I	20 volte al giorno
5	Aggiunta di un account	I	5.000 volte al giorno
6	Rimozione di un account (inclusi i suoi preferiti)	I	500 volte al giorno
7	Inserimento di una valutazione	I	10.000 volte al giorno
8	Aggiunta di un preferito	I	3.000 volte al giorno
9	Inserimento delle informazioni riguardanti una proiezione	I	500 volte al giorno
10	Rimozione delle informazioni riguardanti a una proiezione passata	B	1 volta a settimana
11	Aggiornamento della media dei voti di un contenuto	B	10 volte al giorno

I = INTERATTIVA

B = AUTOMATICA (BATCH)

## 2.3. Ristrutturazione dello schema E-R

### 2.3.1. Analisi delle ridondanze

#### MEDIA DEI VOTI DI UN CONTENUTO

Analizziamo le prestazioni con la presenza e l'assenza della ridondanza e confrontiamo i risultati.

##### Tavola dei volumi

Contenuto	E	50.000 (ipotetica quantità di contenuti)
Valutazione	A	10.000.000 (circa 21 valutazioni per utente)

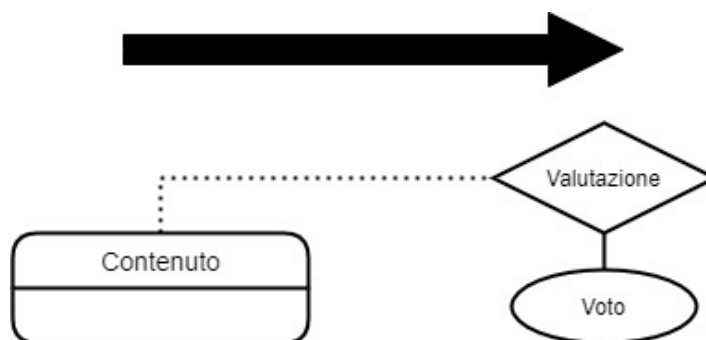
##### Tavola delle operazioni

1	Consulta dati di un Contenuto da parte di un utente	I	50.000 volte al giorno
11	Aggiornamento della media dei voti di un contenuto	B	10 volte al giorno

#### Scenario A: assenza della ridondanza

**OPERAZIONE 1:** Consulta dati di un contenuto da parte di un utente

**Schema dell'operazione**



##### Tavola degli accessi

Concetto	Costrutto	Accesso	Tipo
Contenuto	E	1	Lettura
Valutazione	A	200*	Lettura

\*In media un contenuto avrà 200 valutazioni

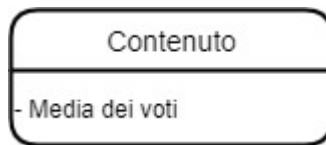
**OPERAZIONE 2:** Aggiornamento della media dei voti di un contenuto

Questa operazione è fattibile solo in presenza della ridondanza.

### Scenario B: presenza della ridondanza

**OPERAZIONE 1:** Consulta dati di un contenuto da parte di un utente

**Schema dell'operazione**

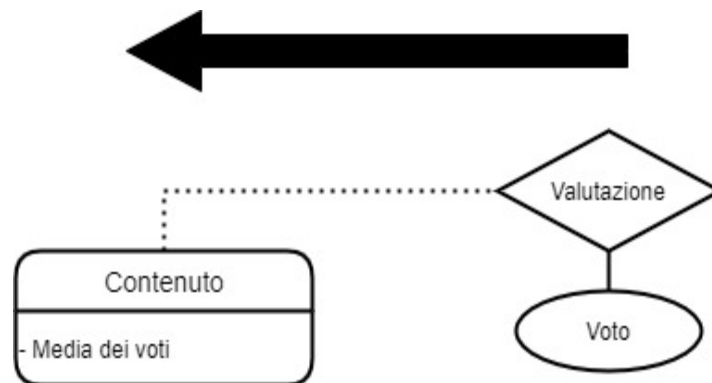


**Tavola degli accessi**

Concetto	Costrutto	Accesso	Tipo
Contenuto	E	1	Lettura

**OPERAZIONE 2:** Aggiornamento della media dei voti di un contenuto

**Schema dell'operazione**



**Tavola degli accessi**

Concetto	Costrutto	Accesso	Tipo
Valutazione	A	200	Lettura
Contenuto	E	1	Scrittura

### Scenario A: assenza della ridondanza

**Costi:**

- Spazio: 0 byte
- Tempo:
  - Operazione 1:  $(1 + 200) * 50.000$  accessi in lettura al giorno, quindi 10.050.000 accessi al giorno.
  - Operazione 2: non eseguibile

## **Scenario B: presenza della ridondanza**

### **Costi:**

- Spazio: La media di un contenuto viene salvata su 1 byte.  $1 \text{ byte} * 50.000 \text{ contenuti} = 50.000 \text{ byte} = 50 \text{ Kbyte}$
- Tempo:
  - Operazione 1:  $(1) * 50.000 \text{ accessi in lettura, quindi } 50.000 \text{ accessi al giorno.}$
  - Operazione 2:  $200 * 50.000 \text{ contenuti} * 10 = 200.000.000 \text{ volte al giorno accessi in lettura e } 1 * 50.000 * 10 \text{ volte di accessi in scrittura al giorno. Contando doppi gli accessi in scrittura, si hanno quindi } 1.000.000 \text{ accessi al giorno.}$

### **Conclusione:**

- Eliminando la ridondanza risparmieremo circa 50 Kbyte, con circa 10.050.000 accessi al giorno.
- Mantenendo la ridondanza occupiamo circa 50 Kbyte ed avremmo circa 190.950.000 accessi in più al giorno.

**Quindi possiamo concludere che mantenere aggiornato il dato ridondante costa molto di più che ricalcolarlo.**

## **2.3.2. Eliminazione delle generalizzazioni**

### **Generalizzazione di account in redattori e utenti**

Abbiamo deciso di accorpare il genitore (account) nei figli, perché le nostre operazioni fanno ampiamente distinzione tra le due entità figlie (redattori e utenti).

### **Generalizzazione di piattaforme in canali TV e streaming video**

Abbiamo deciso di accorpare i figli della generalizzazione nel genitore, perché nel nostro modello le operazioni non fanno troppa distinzione tra le istanze delle varie entità figlie, inoltre dato che non abbiamo attributi specifici per le entità figlie non dobbiamo inserire le business rules, è sufficiente inserire un attributo A-tipo per distinguere le istanze delle due entità.

### **Generalizzazione di sala in sala disponibile e sala non disponibile**

Come nel caso precedente abbiamo deciso di adottare la prima soluzione, ovvero quella di accorpare i figli della generalizzazione nel genitore, perché le operazioni non fanno distinzione tra le istanze delle varie entità figlie, inoltre dato che non abbiamo attributi specifici per le entità figlie non dobbiamo inserire le business rules, sarà sufficiente inserire un attributo A-tipo per distinguere le istanze delle due entità.

### **Generalizzazione di contenuto in film, serie TV e programmi**

Abbiamo deciso di adottare la terza soluzione, di sostituire quindi la generalizzazione con associazioni per legare l'entità genitore alle entità figlie. Perché nel nostro modello abbiamo delle operazioni che fanno distinzione negli accessi alle occorrenze delle entità figlie e quella del genitore.

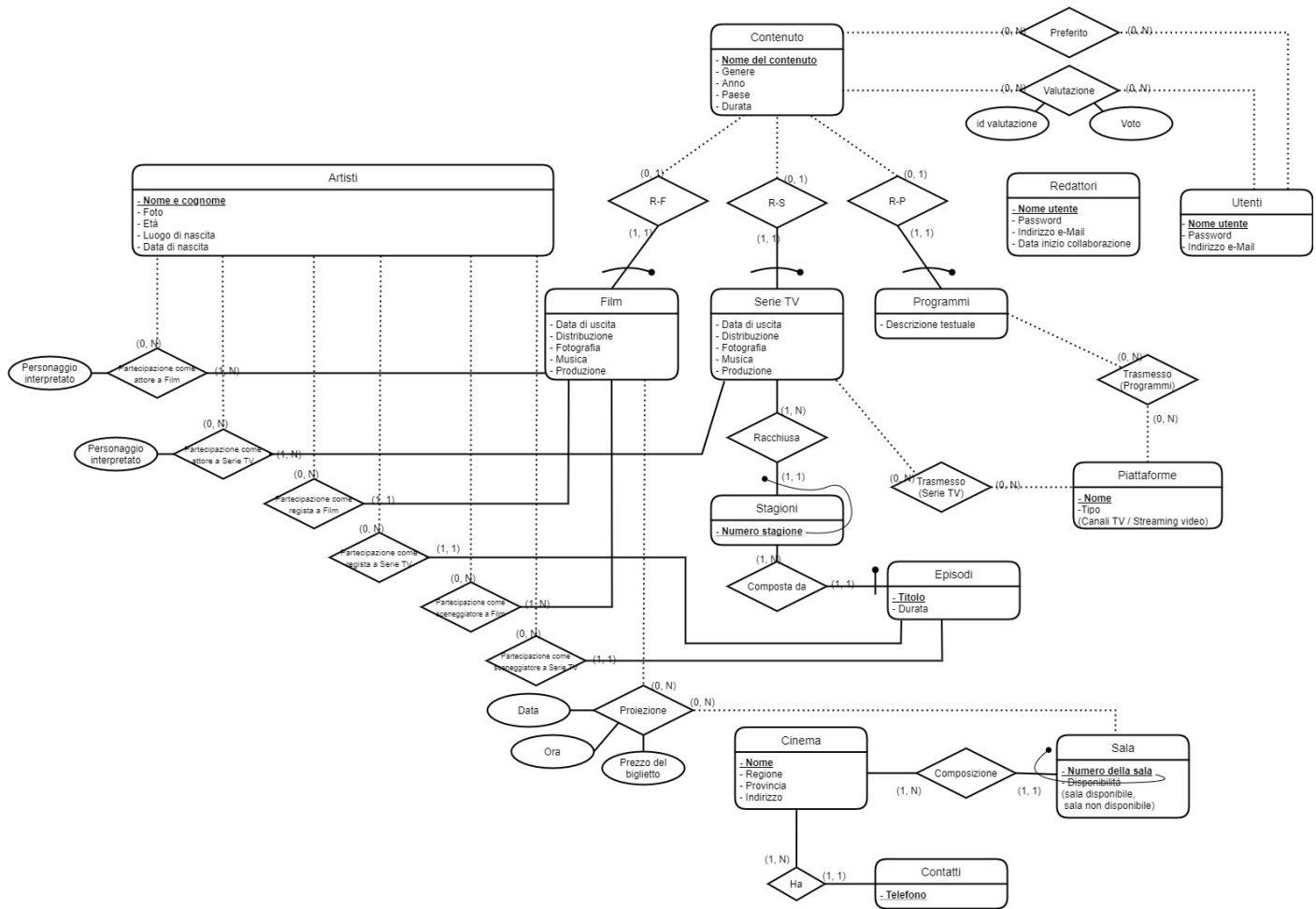
## **2.3.3. Eventuale partizionamento/accorpamento di entità e associazioni**

Non abbiamo ritenuto necessario partizionare o accorpare entità o associazioni nel nostro schema E-R.

## **2.3.4. Eventuale scelta degli identificatori principali**

Abbiamo deciso di introdurre un nuovo identificatore nell'entità Valutazione (id\_valutazione) al fine di identificare univocamente le valutazioni nello schema relazionale. In questo modo, qualora un utente si cancellasse non si creerebbero problemi di consistenza.

## 2.4. Schema E-R ristrutturato + business rules



### Business rule

- 1) I contenuti recenti associati ad un artista sono un sottoinsieme dei contenuti a cui quest'ultimo ha partecipato, i quali contenuti non hanno una data di uscita non più vecchia di un'anno.
- 2) Un artista può partecipare allo stesso contenuto come attore, oppure come regista, oppure come sceneggiatore.
- 3) La media dei voti di un contenuto si riferisce alla media delle valutazioni di quel contenuto all'interno dell'associazione "Valutazione".
- 4) Ogni occorrenza di contenuto non può partecipare contemporaneamente a R-F, R-S e R-P.
- 5) Contenuto deve partecipare ALMENO a una delle relazioni R-F, R-S o R-P.

### 2.4.1. Business rules

- 1) I contenuti recenti associati ad un artista sono un sottoinsieme dei contenuti a cui quest'ultimo ha partecipato, i quali contenuti non hanno una data di uscita non più vecchia di un'anno.
- 2) Un artista può partecipare allo stesso contenuto come attore, oppure come regista, oppure come sceneggiatore.
- 3) La media dei voti di un contenuto si riferisce alla media delle valutazioni di quel contenuto all'interno dell'associazione "Valutazione".
- 4) Ogni occorrenza di contenuto non può partecipare contemporaneamente a R-F, R-S e R-P.
- 5) Contenuto deve partecipare ALMENO a una delle relazioni R-F, R-S o R-P.

## 2.5. Schema relazionale

Utenti(Nome utente, Password, Indirizzo e-mail)

Redattori(Nome utente, Password, Indirizzo e-mail, Data inizio collaborazione)

Valutazione(Utenti, Contenuto, id valutazione, Voto)

Preferito(Utenti, Contenuto)

Contenuto(Nome del contenuto, Genere, Anno, Paese, Durata)

Film(Contenuto, Data di uscita, Distribuzione, Fotografia, Musica, Produzione, Regista)  
Proiezione(Film, Cinema, Numero della sala, Data, Ora, Prezzo del biglietto)

Serie TV(Contenuto, Data di uscita, Distribuzione, Fotografia, Musica, Produzione)  
Trasmesso(Serie TV)(Serie TV, Piattaforme)  
Stagioni(Numero stagione, Serie TV)  
Episodi(Titolo, Numero stagione, Serie TV, Durata, Regista, Sceneggiatore)

Programmi(Contenuto, Descrizione Testuale)  
Trasmesso(Programmi)(Programmi, Piattaforme)

Piattaforme(Nome, Tipo)

Cinema(Nome, Regione, Provincia, Indirizzo)  
Sala(Cinema, Numero della sala, Disponibilità)  
Contatti(Telefono, Cinema)

Artisti(Nome e cognome, Foto, Età, Luogo di nascita, Data di nascita)

PAF(Artisti, Film, Personaggio interpretato)  
PAS(Artisti, Serie TV, Personaggio interpretato)  
PSF(Artisti, Film)

Note:

- \*PAF si intende come "Partecipazione come attore a Film"
- \*PAS si intende come "Partecipazione come attore a Serie TV"
- \*PSF si intende come "Partecipazione come sceneggiatore a Film"

### 2.5.1. Vincoli di integrità referenziale

- Valutazione(Utenti) referencia Utenti(Nome utente)
- Valutazione(Contenuto) referencia Contenuto(Nome del contenuto)
- Preferiti(Utenti) referencia Utenti(Nome utente)
- Preferiti(Contenuto) referencia Contenuto(Nome del contenuto)
- Film(Contenuto) referencia Contenuto(Nome del contenuto)
- Film(Regista) referencia Artisti(Nome e cognome)
- Proiezione(Film) referencia Film(Contenuto)
- Proiezione(Numero della sala, Cinema) referencia Sala(Numero della sala, Cinema)
- Serie TV(Contenuto) referencia Contenuto(Nome del contenuto)
- Trasmesso(Serie TV)(Serie TV) referencia Serie TV(Contenuto)
- Trasmesso(Serie TV)(Piattaforme) referencia Piattaforme(Nome)
- Stagione(Serie TV) referencia SerieTV(Contenuto)
- Episodi(Numero stagione, Serie TV) referencia Stagioni(Numero stagione, Serie TV)
- Episodi(Regista) referencia Artisti(Nome e cognome)
- Episodi(Sceneggiatore) referencia Artisti(Nome e cognome)
- Programmi(Contenuto) referencia Contenuto(Nome del contenuto)



- Trasmesso(Programmi)(Programmi) referenzia Programmi(Contenuto)
- Trasmesso(Programmi)(Piattaforme) referenzia Piattaforme(Nome)
- Sala(Cinema) referenzia Cinema(Nome)
- Contatti(Cinema) referenzia Cinema(Nome)
- PAF(Artisti) referenzia Artisti(Nome e cognome)
- PAF(Film) referenzia Film(Contenuto)
- PAS(Artisti) referenzia Artisti(Nome e cognome)
- PAS(Serie TV) referenzia SerieTV(Contenuto)
- PSF(Artisti) referenzia Artisti(Nome e cognome)
- PSF(Film) referenzia Film(Contenuto)

## 3. Implementazione

---

### 3.1 DDL di creazione del database

```
begin;
```

```
drop table if exists Utenti;  
drop table if exists Redattori;  
drop table if exists Valutazione;  
drop table if exists Preferito;  
drop table if exists Contenuto;  
drop table if exists Film;  
drop table if exists Proiezione;  
drop table if exists SerieTV;  
drop table if exists Trasmesso;  
drop table if exists Stagioni;  
drop table if exists Episodi;  
drop table if exists Programmi;  
drop table if exists Trasmesso;  
drop table if exists Piattaforme;  
drop table if exists Cinema;  
drop table if exists Sala;  
drop table if exists Contatti;  
drop table if exists Artisti;  
drop table if exists PAF;  
drop table if exists PAS;  
drop table if exists PSF;  
drop table if exists PSS;
```

```
/*Utenti(Nome utente, Password, Indirizzo e-mail)*/
```

```
create table Utenti (  
    Nome_utente varchar(30),  
    Password varchar(50) not null,  
    Indirizzo_email varchar(100) not null,  
    primary key(Nome_utente)  
);
```

```
/*Redattori(Nome utente, Password, Indirizzo e-mail, Data inizio collaborazione)*/
```

```
create table Redattori (  
    Nome_utente varchar(30),  
    Password varchar(50) not null,  
    Indirizzo_email varchar(100) not null,  
    Data_inizio_collaborazione date not null,  
    primary key(Nome_utente)  
);
```

```
/*Contenuto(Nome del contenuto, Genere, Anno, Paese, Durata)*/
```

```
create table Contenuto (  
    Nome_del_contenuto varchar(50),  
    Genere varchar(20) not null,  
    Anno numeric(4) not null,  
    Paese varchar(20) not null,  
    Durata numeric(3) not null check (Durata > 0),  
    primary key(Nome_del_contenuto)  
);
```

```
/*Artisti(Nome e cognome, Foto, Età, Luogo di nascita, Data di nascita)*/
```

```
create table Artisti (  
    Nome_e_cognome varchar(50) not null,  
    Foto bytea,  
    Età numeric(3) not null,  
    Luogo_di_nascita varchar(20) not null,  
    Data_di_nascita date not null,  
    primary key(Nome_e_cognome)  
);
```

```
/*Valutazione(Utenti, Contenuto, id valutazione, Voto)*/
```

```
/*Valutazione(Utenti) referenza Utenti(Nome utente)
```

```
Valutazione(Contenuto) referenzia Contenuto(Nome del contenuto)*/
```

```
create table Valutazione (  
    Utenti varchar(20),  
    Contenuto varchar(50),  
    id_valutazione SERIAL,  
    Voto numeric(1) not null check (Voto >= 0 and Voto <= 5),  
    primary key(id_valutazione),  
    foreign key(Utenti) references Utenti(Nome_utente) on delete set null,  
    foreign key (Contenuto) references Contenuto(Nome_del_contenuto) on delete  
cascade on update cascade  
);
```

```
/*Preferito(Utenti, Contenuto)*/
```

```
/*Preferiti(Utenti) referenza Utenti(Nome utente)
```

```
Preferiti(Contenuto) referenzia Contenuto(Nome del contenuto)*/
```

```
create table Preferito (  
    Utenti varchar(20),  
    Contenuto varchar(50),  
    primary key(Utenti, Contenuto),  
    foreign key(Utenti) references Utenti(Nome_utente) on delete cascade on update  
cascade,  
    foreign key(Contenuto) references Contenuto(Nome_del_contenuto) on delete cascade  
on update cascade  
);
```

```

/*Film(Contenuto, Data di uscita, Distribuzione, Fotografia, Musica, Produzione,
Regista)*/
/*Film(Contenuto) referenzia Contenuto(Nome del contenuto)
Film(Regista) referenzia Artisti(Nome e cognome)*/

create table Film (
    Contenuto varchar(50),
    Data_di_uscita date not null,
    Distribuzione varchar(30) not null,
    Fotografia varchar(50) not null,
    Musica varchar(50) not null,
    Produzione varchar(50) not null,
    Regista varchar(50),
    primary key(Contenuto),
    foreign key(Contenuto) references Contenuto(Nome_del_contenuto) on delete cascade
on update cascade,
    foreign key(Regista) references Artisti(Nome_e_cognome) on delete set null on
update cascade
);

/*Cinema(Nome, Regione, Provincia, Indirizzo)*/

create table Cinema (
    Nome varchar(50),
    Regione varchar(20) not null,
    Provincia char(2) not null,
    Indirizzo varchar(100) not null,
    primary key(Nome)
);

/*Sala(Cinema, Numero della sala, Disponibilità)*/
/*Sala(Cinema) referenzia Cinema(Nome)*/

create table Sala (
    Cinema varchar(50),
    Numero_della_sala numeric(2),
    Disponibilità boolean not null,
    primary key(Cinema, Numero_della_sala),
    foreign key(Cinema) references Cinema(Nome) on update cascade on delete cascade
);

/*Proiezione(Film, Cinema, Numero della sala, Data, Ora, Prezzo del biglietto)*/
/*Proiezione(Film) referenzia Film(Contenuto)
Proiezione(Numero della sala, Cinema) referenzia Sala(Numero della sala, Cinema)*/

create table Proiezione (
    Film varchar(50),

```

```

Cinema varchar(50),
Numero_della_sala numeric(2),
Data date not null,
Ora time not null,
Prezzo_del_biglietto decimal(4,2) not null,
primary key(Film, Cinema, Numero_della_sala, Data, Ora),
foreign key(Film) references Film(Contenuto) on delete cascade on update cascade,
foreign key(Numero_della_sala, Cinema) references Sala(Numero_della_sala, Cinema)
on delete cascade on update cascade
);

```

```

/*Serie_TV(Contenuto, Data di uscita, Distribuzione, Fotografia, Musica,
Produzione)*/

```

```

/*Serie_TV(Contenuto) referencia Contenuto(Nome del contenuto)*/

```

```

create table SerieTV (
    Contenuto varchar(50),
    Data_di_uscita date not null,
    Distribuzione varchar(30) not null,
    Fotografia varchar(50) not null,
    Musica varchar(50) not null,
    Produzione varchar(50) not null,
    primary key(Contenuto),
    foreign key(Contenuto) references Contenuto(Nome_del_contenuto) on delete cascade
on update cascade
);

```

```

/*Piattaforme(Nome, Tipo)*/

```

```

create table Piattaforme (
    Nome varchar(50),
    Tipo varchar(20) not null,
    primary key(Nome)
);

```

```

/*Trasmesso(Serie TV)(Serie TV, Piattaforme)*/

```

```

/*Trasmesso(Serie TV)(Serie TV) referencia Serie TV(Contenuto)
Trasmesso(Serie TV)(Piattaforme) referencia Piattaforme(Nome)*/

```

```

create table Trasmesso_SerieTV (
    SerieTV varchar(50),
    Piattaforme varchar(50),
    primary key(SerieTV, Piattaforme),
    foreign key(SerieTV) references SerieTV(Contenuto) on delete cascade on update
cascade,
    foreign key(Piattaforme) references Piattaforme(Nome) on delete cascade on update
cascade
);

```

```

/*Stagioni(Numero stagione, Serie TV)*/
/*Stagione(Serie TV) referencia SerieTV(Contenuto)*/

create table Stagioni (
    Numero_stagione numeric(2) check (Numero_stagione > 0),
    SerieTV varchar(50),
    primary key(Numero_stagione, SerieTV),
    foreign key(SerieTV) references SerieTV(Contenuto) on delete cascade on update
cascade
);

/*Episodi(Titolo, Numero stagione, Serie TV, Durata, Regista)*/
/*Episodi(Numero stagione, Serie TV) referencia Stagioni(Numero stagione, Serie TV)
Episodi(Regista) referencia Artisti(Nome e cognome)*/

create table Episodi (
    Titolo varchar(50),
    Numero_stagione numeric(2) check(Numero_stagione > 0),
    SerieTV varchar(50),
    Durata numeric(3) not null check(Durata > 0),
    Regista varchar(50),
    Sceneggiatore varchar(50),
    primary key(Titolo, Numero_stagione, SerieTV),
    foreign key(Numero_stagione, SerieTV) references Stagioni(Numero_stagione,
SerieTV) on delete cascade on update cascade,
    foreign key(Regista) references Artisti(Nome_e_cognome) on delete set null on
update cascade,
    foreign key(Sceneggiatore) references Artisti(Nome_e_cognome) on delete set null
on update cascade
);

/*Programmi(Contenuto, Descrizione Testuale)*/
/*Programmi(Contenuto) referencia Contenuto(Nome del contenuto)*/

create table Programmi (
    Contenuto varchar(50),
    Descrizione_testuale varchar(1000),
    primary key(Contenuto),
    foreign key(contenuto) references Contenuto(Nome_del_contenuto) on delete cascade
on update cascade
);

/*Trasmesso(Programmi)(Programmi, Piattaforme)*/
/*Trasmesso(Programmi)(Programmi) referencia Programmi(Contenuto)
Trasmesso(Programmi)(Piattaforme) referencia Piattaforme(Nome)*/

create table Trasmesso_Programmi (
    Programmi varchar(50),
    Piattaforme varchar(50),

```



```
primary key(Programmi, Piattaforme),
foreign key(Programmi) references Programmi(Contenuto) on update cascade on
delete cascade,
foreign key(Piattaforme) references Piattaforme(Nome) on update cascade on delete
cascade
);
```

```
/*Contatti(Telefono, Cinema)*/
/*Contatti(Cinema) referenzia Cinema(Nome)*/
```

```
create table Contatti (
Telefono numeric(10),
Cinema varchar(50),
primary key(Telefono),
foreign key(Cinema) references Cinema(Nome) on update cascade on delete cascade
);
```

```
/*PAF(Artisti, Film, Personaggio interpretato)*/
/*PAF(Artisti) referenzia Artisti(Nome e cognome)
PAF(Film) referenzia Film(Contenuto)*/
```

```
create table PAF (
Artisti varchar(50),
Film varchar(50),
Personaggio_interpretato varchar(50) not null,
primary key(Artisti, Film),
foreign key(Artisti) references Artisti(Nome_e_cognome) on update cascade on
delete cascade,
foreign key(Film) references Film(Contenuto) on update cascade on delete cascade
);
```

```
/*PAS(Artisti, Serie TV, Personaggio interpretato)*/
/*PAS(Artisti) referenzia Artisti(Nome e cognome)
PAS(Serie TV) referenzia SerieTV(Contenuto)*/
```

```
create table PAS (
Artisti varchar(50),
SerieTV varchar(50),
Personaggio_interpretato varchar(50) not null,
primary key(Artisti, SerieTV),
foreign key(Artisti) references Artisti(Nome_e_cognome) on update cascade on
delete cascade,
foreign key(SerieTV) references SerieTV(Contenuto) on update cascade on delete
cascade
);
```

```
/*PSF(Artisti, Film)*/
/*PSF(Artisti) referenzia Artisti(Nome e cognome)
```

```
PSF(Film) referencia Film(Contenuto)*/
```

```
create table PSF (  
    Artisti varchar(50),  
    Film varchar(20),  
    primary key(Artisti, Film),  
    foreign key(Artisti) references Artisti(Nome_e_cognome) on update cascade on  
delete cascade,  
    foreign key(Film) references Film(Contenuto) on update cascade on delete cascade  
);  
  
commit;
```

## 3.2 DML di popolazione del database

```
/*Utenti(Nome utente, Password, Indirizzo e-mail)*/
```

```
begin;  
insert into Utenti values('Utente1', 'password', 'utente_1@gmail.com');  
insert into Utenti values('Utente2', 'qwerty', 'utente_2@edu.unito.com');  
insert into Utenti values('Utente3', '123', 'utente_3@libero.com');  
commit;
```

```
/*Redattori(Nome utente, Password, Indirizzo e-mail, Data inizio  
collaborazione)*/
```

```
begin;  
insert into Redattori values('Redattore1', 'password', 'redazione_1@gmail.com',  
DATE '2015-12-17');  
insert into Redattori values('Redattore2', 'qwerty', 'redazione_2@gmail.com',  
DATE '2016-06-05');  
insert into Redattori values('Redattore3', '123', 'redazione_3@gmail.com', DATE  
'2017-10-28');  
commit;
```

```
/*Contenuto(Nome del contenuto, Genere, Anno, Paese, Durata)*/
```

```
begin;  
insert into Contenuto values('Film1', 'Drammatico', 1980, 'Inghilterra', 195);  
insert into Contenuto values('Film2', 'Avventura', 1999, 'Spagna', 130);  
insert into Contenuto values('SerieTv1', 'Avventura', 2004, 'Inghilterra', 45);  
insert into Contenuto values('SerieTv2', 'Avventura', 2015, 'Francia', 30);  
insert into Contenuto values('Programma1', 'Intrattenimento', 2010 , 'Italia',  
33);  
insert into Contenuto values('Programma2', 'Intrattenimento', 2013 , 'Italia',  
50);  
commit;
```

```
/*Artisti(Nome e cognome, Foto, Età, Luogo di nascita, Data di nascita)*/
```

```
begin;
```

```

insert into Artisti(Nome_e_cognome, Età, Luogo_di_nascita, Data_di_nascita)
values('Artista1', 15, 'Torino', DATE '2000-07-11');
insert into Artisti(Nome_e_cognome, Età, Luogo_di_nascita, Data_di_nascita)
values('Artista2', 24, 'New York', DATE '1996-07-22');
insert into Artisti(Nome_e_cognome, Età, Luogo_di_nascita, Data_di_nascita)
values('Artista3', 33, 'Roma', DATE '1988-08-23');
insert into Artisti(Nome_e_cognome, Età, Luogo_di_nascita, Data_di_nascita)
values('Artista4', 42, 'Milano', DATE '1978-09-07');
insert into Artisti(Nome_e_cognome, Età, Luogo_di_nascita, Data_di_nascita)
values('Artista5', 51, 'Roma', DATE '1969-10-22');
insert into Artisti(Nome_e_cognome, Età, Luogo_di_nascita, Data_di_nascita)
values('Artista6', 45, 'Los Angeles', DATE '1974-11-11');
insert into Artisti(Nome_e_cognome, Età, Luogo_di_nascita, Data_di_nascita)
values('Artista7', 51, 'Londra', DATE '1969-01-17');
insert into Artisti(Nome_e_cognome, Età, Luogo_di_nascita, Data_di_nascita)
values('Artista8', 54, 'Roma', DATE '1965-06-03');
insert into Artisti(Nome_e_cognome, Età, Luogo_di_nascita, Data_di_nascita)
values('Artista9', 42, 'Verona', DATE '1958-09-07');
insert into Artisti(Nome_e_cognome, Età, Luogo_di_nascita, Data_di_nascita)
values('Artista10', 42, 'Napoli', DATE '1967-10-23');
commit;

```

```

/*Valutazione(Utenti, Contenuto, id valutazione, Voto)*/
begin;
insert into Valutazione(Utenti, Contenuto, Voto) values('Utente1', 'Film1', 5);
insert into Valutazione(Utenti, Contenuto, Voto) values('Utente1', 'SerieTv1',
3);
insert into Valutazione(Utenti, Contenuto, Voto) values('Utente1',
'Programma1', 2);
insert into Valutazione(Utenti, Contenuto, Voto) values('Utente2', 'Film2', 1);
insert into Valutazione(Utenti, Contenuto, Voto) values('Utente2', 'SerieTv2',
2);
insert into Valutazione(Utenti, Contenuto, Voto) values('Utente2',
'Programma2', 4);
insert into Valutazione(Utenti, Contenuto, Voto) values('Utente3', 'Film1', 3);
insert into Valutazione(Utenti, Contenuto, Voto) values('Utente3', 'SerieTv1',
2);
insert into Valutazione(Utenti, Contenuto, Voto) values('Utente3',
'Programma1', 1);
commit;

```

```

/*Preferito(Utenti, Contenuto)*/
begin;
insert into Preferito values('Utente1', 'Film1');
insert into Preferito values('Utente2', 'SerieTv1');
insert into Preferito values('Utente3', 'Programma1');
commit;

```

```

/*Film(Contenuto, Data di uscita, Distribuzione, Fotografia, Musica,
Produzione, Regista)*/
begin;
insert into Film values('Film1', DATE '1980-01-16', 'distribuzione_1',
'fotografia_1', 'musica_1', 'produzione_1', 'Artista1');
insert into Film values('Film2', DATE '1999-01-16', 'distribuzione_2',
'fotografia_2', 'musica_2', 'produzione_2', 'Artista2');
commit;

/*Cinema(Nome, Regione, Provincia, Indirizzo)*/
begin;
insert into Cinema values('Cinema1', 'Piemonte', 'TO', 'Corso Vittorio
Emanuele, 55');
insert into Cinema values('Cinema2', 'Piemonte', 'CN', 'Via Torino, 55');
commit;

/*Sala(Cinema, Numero della sala, Disponibilità)*/
begin;
insert into Sala values('Cinema1', 1, TRUE);
insert into Sala values('Cinema1', 2, FALSE);
insert into Sala values('Cinema1', 3, FALSE);
insert into Sala values('Cinema1', 4, FALSE);
insert into Sala values('Cinema2', 1, TRUE);
insert into Sala values('Cinema2', 2, FALSE);
insert into Sala values('Cinema2', 3, FALSE);
insert into Sala values('Cinema2', 4, FALSE);
commit;

/*Proiezione(Film, Cinema, Numero della sala, Data, Ora, Prezzo del
biglietto)*/
begin;
insert into Proiezione values('Film1', 'Cinema1', 1, DATE '2020-06-03', TIME
'19:00', 10.50);
insert into Proiezione values('Film1', 'Cinema2', 2, DATE '2020-10-15', TIME
'23:00', 9.00);
commit;

/*SerieTV(Contenuto, Data di uscita, Distribuzione, Fotografia, Musica,
Produzione)*/
begin;
insert into SerieTV values('SerieTv1', DATE '2004-09-22', 'distribuzione_1',
'fotografia_1', 'musica_1', 'produzione_1');
insert into SerieTV values('SerieTv2', DATE '2015-10-05', 'distribuzione_2',
'fotografia_2', 'musica_2', 'produzione_2');
commit;

```

```

/*Piattaforme(Nome, Tipo)*/
begin;
insert into Piattaforme values('TantiFilm', 'sito streaming');
insert into Piattaforme values('CinemaGratis', 'sito streaming');
insert into Piattaforme values('CinemaHD', 'sito streaming');
insert into Piattaforme values('FilmStreamingIta', 'sito streaming');
insert into Piattaforme values('AltaDefinizione', 'sito streaming');
insert into Piattaforme values('Rai1', 'canale TV');
insert into Piattaforme values('Rai2', 'canale TV');
insert into Piattaforme values('Rai3', 'canale TV');
insert into Piattaforme values('Rete4', 'canale TV');
insert into Piattaforme values('Canale5', 'canale TV');
insert into Piattaforme values('ItaliaUno', 'canale TV');
commit;

```

```

/*Trasmesso(Serie TV)(Serie TV, Piattaforme)*/
begin;
insert into Trasmesso_SerieTV values('SerieTv1', 'Rai1');
insert into Trasmesso_SerieTV values('SerieTv1', 'TantiFilm');
insert into Trasmesso_SerieTV values('SerieTv2', 'ItaliaUno');
commit;

```

```

/*Stagioni(Numero stagione, Serie TV)*/
begin;
insert into Stagioni values(1, 'SerieTv1');
insert into Stagioni values(2, 'SerieTv1');
insert into Stagioni values(1, 'SerieTv2');
insert into Stagioni values(2, 'SerieTv2');
commit;

```

```

/*Episodi(Titolo, Numero stagione, Serie TV, Durata, Regista, Sceneggiatore)*/
begin;
insert into Episodi values('Episodio1', 1, 'SerieTv1', 30, 'Artista1',
'Artista8');
insert into Episodi values('Episodio2', 1, 'SerieTv1', 44, 'Artista2',
'Artista7');
insert into Episodi values('Episodio3', 2, 'SerieTv1', 25, 'Artista3',
'Artista6');
insert into Episodi values('Episodio4', 2, 'SerieTv1', 48, 'Artista4',
'Artista5');
insert into Episodi values('Episodio1', 1, 'SerieTv2', 20, 'Artista5',
'Artista4');
insert into Episodi values('Episodio2', 1, 'SerieTv2', 54, 'Artista6',
'Artista3');

```

```
insert into Episodi values('Episodio3', 2, 'SerieTv2', 75, 'Artista7',
'Artista2');
insert into Episodi values('Episodio4', 2, 'SerieTv2', 68, 'Artista8',
'Artista1');
commit;
```

```
/*Programmi(Contenuto, Descrizione Testuale)*/
begin;
insert into Programmi values('Programma1', 'Sei concorrenti si sfidano in
questo classico gioco a premi ad eliminazione che prevede sette diverse
sfide.');
```

```
insert into Programmi values('Programma2', 'E un programma televisivo italiano
ideato da Antonio Ricci, in onda dal lunedì al sabato in access prime time su
Canale 5 sin dal 2015. La prima edizione, del 2015, andò in onda su Italia 1');
commit;
```

```
/*Trasmesso(Programmi)(Programmi, Piattaforme)*/
begin;
insert into Trasmesso_Programmi values('Programma1', 'Rai1');
insert into Trasmesso_Programmi values('Programma1', 'Rai2');
insert into Trasmesso_Programmi values('Programma1', 'Rai3');
insert into Trasmesso_Programmi values('Programma2', 'TantiFilm');
insert into Trasmesso_Programmi values('Programma2', 'CinemaGratis');
insert into Trasmesso_Programmi values('Programma2', 'CinemaHD');
commit;
```

```
/*Contatti(Telefono, Cinema)*/
begin;
insert into Contatti values('1234567890', 'Cinema1');
insert into Contatti values('0987654321', 'Cinema1');
insert into Contatti values('1203120122', 'Cinema2');
insert into Contatti values('1345545322', 'Cinema2');
commit;
```

```
/*PAF(Artisti, Film, Personaggio interpretato)*/
begin;
insert into PAF values('Artista1', 'Film1', 'Personaggio interpretato 1');
insert into PAF values('Artista2', 'Film1', 'Personaggio interpretato 2');
insert into PAF values('Artista3', 'Film1', 'Personaggio interpretato 3');
insert into PAF values('Artista4', 'Film1', 'Personaggio interpretato 4');
insert into PAF values('Artista5', 'Film1', 'Personaggio interpretato 5');
insert into PAF values('Artista6', 'Film2', 'Personaggio interpretato 1');
insert into PAF values('Artista7', 'Film2', 'Personaggio interpretato 2');
insert into PAF values('Artista8', 'Film2', 'Personaggio interpretato 3');
insert into PAF values('Artista9', 'Film2', 'Personaggio interpretato 4');
```



```

insert into PAF values('Artista10', 'Film2', 'Personaggio interpretato 5');
commit;

/*PAS(Artisti, Serie TV, Personaggio interpretato)*/
begin;
insert into PAS values('Artista1', 'SerieTv1', 'Personaggio interpretato 1');
insert into PAS values('Artista2', 'SerieTv1', 'Personaggio interpretato 2');
insert into PAS values('Artista3', 'SerieTv1', 'Personaggio interpretato 3');
insert into PAS values('Artista4', 'SerieTv1', 'Personaggio interpretato 4');
insert into PAS values('Artista5', 'SerieTv1', 'Personaggio interpretato 5');
insert into PAS values('Artista6', 'SerieTv2', 'Personaggio interpretato 1');
insert into PAS values('Artista7', 'SerieTv2', 'Personaggio interpretato 2');
insert into PAS values('Artista8', 'SerieTv2', 'Personaggio interpretato 3');
insert into PAS values('Artista9', 'SerieTv2', 'Personaggio interpretato 4');
insert into PAS values('Artista10', 'SerieTv2', 'Personaggio interpretato 5');
commit;

/*PSF(Artisti, Film)*/
begin;
insert into PSF values('Artista1', 'Film1');
insert into PSF values('Artista2', 'Film1');
insert into PSF values('Artista3', 'Film1');
insert into PSF values('Artista4', 'Film2');
insert into PSF values('Artista5', 'Film2');
insert into PSF values('Artista6', 'Film2');
commit;

```

### 3.3 Qualche operazione di cancellazione e modifica per verificare i vincoli e gli effetti causati da operazioni su chiavi esterne

#### **Cancella dalla tabella Utenti l'utente con Nome\_utente uguale a 'Utente1'**

```

begin;
delete from Utenti where Nome_utente = 'Utente1';
commit;

```

#### **Aggiunta di un nuovo utente nella tabella Utenti**

```

begin;
insert into Utenti values('Utente1', 'password', 'utente_1@gmail.com');
commit;

```

#### **Aggiunta di un utente già presente nella tabella Utenti: deve dare errore**

```

begin;
insert into Utenti values('Utente1', 'password', 'utente_1@gmail.com');
commit;

```

**Cancella dalla tabella Contenuto il contenuto con nome 'Film1', devono cancellarsi anche tutti i riferimenti nelle tabelle (Film, Proiezione, ...) al film cancellato**

```
begin;
delete from Contenuto where Nome_del_contenuto = 'Film1';
commit;
```

**Mostra tutte le informazioni riferite alle proiezioni, non dovrebbero comparire le proiezioni riferite al film 'Film1'**

```
begin;
select * from Proiezione
commit;
```

**Aggiunta di una nuova valutazione da parte di un utente**

```
begin;
insert into Valutazione(Utenti, Contenuto, Voto) values('Utente1', 'SerieTv1', 5);
commit;
```

**Cancellazione di un utente, le sue valutazioni non si devono cancellare**

```
begin;
delete from Utenti where Nome_utente = 'Utente2';
commit;
select * from Valutazione;
```

**Cancellazione di un contenuto, le valutazioni connesse a quel contenuto devono cancellarsi**

```
begin;
delete from contenuto where nome_del_contenuto = 'Film2';
commit;
select * from Valutazione;
```

**Cancella dalla tabella Cinema il cinema con nome 'Cinema1', devono cancellarsi anche le sale nella tabella Sale che si riferiscono a quel cinema**

```
begin;
delete from Cinema where Nome = 'Cinema1';
commit;
select * from Sala;
```

**Aggiunta di una sala già presente nella tabella: deve dare errore**

```
begin;
insert into Sala values('Cinema1', 2, TRUE);
commit;
```

**Cancella dalla tabella Piattaforme la piattaforma con nome 'TantiFilm'**

```
begin;
delete from Piattaforme where Nome = 'AltaDefinizione';
commit;
select * from Piattaforme;
```

**Aggiunta di una piattaforma già presente nella tabella: deve dare errore**

```
begin;
insert into Piattaforme values('Rai1', 'canale TV');
commit;
```

**Aggiunta di una serie TV già presente nella tabella: deve dare errore**

```
begin;
```

```
insert into SerieTV values('SerieTv1', DATE '2004-09-22', 'Fox', 'Damon Lindelof',  
'Damon Lindelof', 'Carlton Cuse');  
commit;
```

**Cancella dai contenuti la serie di nome 'SerieTv1', devono cancellarsi anche tutti i riferimenti nelle tabelle (SerieTV, Stagioni, Episodi, ...) a quel contenuto eliminato**

```
begin;  
delete from Contenuto where Nome_del_contenuto = 'SerieTv1';  
commit;  
select * from SerieTV;  
select * from Stagioni;  
select * from Episodi;
```

**Aggiunta di un contatto già esistente nella tabella Contatti: deve dare errore**

```
begin;  
insert into Contatti values('1234567890', 'Cinema1');  
commit;
```

**Cancella dalla tabella Cinema il cinema di nome 'Cinema1', si devono anche cancellare tutti i contatti associati a quel cinema**

```
begin;  
delete from Cinema where Nome = 'Cinema1';  
commit;  
select * from Contatti;
```

**Visualizza la media dei voti di un contenuto**

```
select Contenuto, avg(Voto) Media_dei_voti  
from valutazione  
group by Contenuto
```

**Cancella dalla tabella Artisti un artista che è un regista di un film, il film non deve cancellarsi ma deve semplicemente settarsi a null il regista di quel film**

```
begin;  
delete from Artisti where Nome_e_cognome = 'Artista1';  
commit;  
select * from Film; /*Film1 deve avere regista a null*/
```

**Cancella dalla tabella Artisti un artista che è un regista di un episodio, l'episodio non deve cancellarsi ma deve semplicemente settarsi a null il regista di quell'episodio**

```
begin;  
delete from Artisti where Nome_e_cognome = 'Artista1';  
commit;  
select * from episodi; /*episodio1 deve avere regista a null*/
```

**Cancella dalla tabella Artisti un artista che è uno sceneggiatore di un episodio, l'episodio non deve cancellarsi ma deve semplicemente settarsi a null lo sceneggiatore di quell'episodio**

```
begin;  
delete from Artisti where Nome_e_cognome = 'Artista8';  
commit;  
select * from episodi; /*episodio1 deve avere regista a null*/
```