

PROGETTO DATABASE 2019-20 MAURIZIO KORRI

MATR:902438

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. 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.** 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. **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
Utente	Utente del sito, può essere Redattore	Visitatore Del Sito	Contenuto, Voto
Contenuto	Contenuto del sito, può essere un Film, una Serie o un programma TV	Serial	Piattaforma, Attore, Regista, Cinema, Utente, Stagione, Voto, Utente
Attore	Attore che interpreta un personaggio in una Serie TV o in un Film	Interprete	Film, Serie TV
Regista	Colui che ha la responsabilità operativa di un contenuto		Film, Episodio, Programma TV
Piattaforma	Luogo in cui vanno in onda Serie o programmi TV. Possono		Programma TV, Serie TV

	essere Canali TV o Piattaforme di Streaming		
Stagione	Componente di una serie TV, e' composta da Episodi		Serie TV, Episodio
Episodio	Componente di una Stagione di Una serie TV.		Stagione, Regia
Cinema	Luogo in cui vengono mostrati i film, composto da Sale		Sala
Sala	Sala in cui viene proiettato un Film, fa parte di un cinema		Cinema, Film
Cinema	Posto in cui vengono proiettati i film		Sala
Proiezione	L'attuale proiezione del film		Sala, Film

1.3 Requisiti rivisti e strutturati in gruppi di frasi omogenee.

Frase di Carattere Generale:

Si vuole realizzare una base di dati per la gestione di un sito che fornisce informazioni su film in uscita nei cinema, e su serie e programmi TV in onda.

Frase relative agli Utenti:

Gli utenti sono identificati dal Nome utente o dalla E-Mail.

Degli utenti rappresentiamo nome utente, password e indirizzo email.

Gli utenti possono lasciare un voto (da 1 a 5 stelline) su un contenuto.

Gli utenti possono inserire tra i preferiti un contenuto.

Alla cancellazione di un utente dal sito, i suoi preferiti vengono rimossi, mentre i suoi voti espressi rimangono in memoria.

Gli utenti possono essere redattori.

Frase relative ai tipi specifici di Utente (Redattori):

Dei redattori rappresentiamo la data di inizio collaborazione.

I redattori si occupano di inserire i dati di film e serie, comprese le date di uscita e programmazione in sala e in TV.

Frase relative ai Contenuti:

I contenuti sono identificati dal nome.

Dei contenuti rappresentiamo nome, paese, genere, produzione, musica, sceneggiatura, fotografia.

I contenuti sono divisi in: film, serie TV, programmi TV.

Le Informazioni relative ai contenuti vengono aggiornate dai redattori.

Frase relative ai tipi specifici di Contenuti:

Dei contenuti che sono Film rappresentiamo:

data di uscita, anno, durata, distribuzione, sceneggiatore (con relativi nome, cognome), e la media del voto dato dagli utenti.

A ogni film è associata una lista di attori con l'indicazione del personaggio interpretato.

A ogni film è associata la sala del cinema in cui verrà proiettato, con prezzo del biglietto.

A ogni film è associata la regia del film stesso.

Dei contenuti che sono Serie TV rappresentiamo:

Distribuzione, anno, data uscita.

A ogni serie TV è associata la composizione in stagioni (al quale è associata la composizione in episodi).

A ogni serie TV è associata una lista di attori con l'indicazione del personaggio interpretato.

A ogni serie TV è associata la piattaforma in cui verrà trasmessa.

Dei contenuti che sono Programmi TV rappresentiamo:

La media del voto data dagli utenti, una descrizione testuale, e la durata.

A ogni programma TV è associata la piattaforma in cui verrà trasmesso in cui viene rappresentato l'anno di messa in onda.

A ogni programma TV è associata la regia del programma stesso.

Frase Relative alle Stagioni:

Le stagioni sono identificate dalla serie TV al quale fanno parte e dal numero della stagione.

Delle stagioni rappresentiamo il numero stagione, la trama e delle curiosità'.

Ad ogni stagione è associata la composizione in episodi.

Frase Relative agli Episodi:

Gli episodi sono identificati dalla stagione al quale fanno parte e dal numero dell'episodio.

Degli episodi rappresentiamo il numero dell'episodio, la durata, il nome, lo sceneggiatore (con relativo nome e cognome).

Ad ogni episodio è associata la regia dell'episodio stesso.

Frase Relative agli Attori:

Gli attori sono identificati dal nome e cognome.

Degli attori rappresentiamo nome, cognome, età, data di nascita, luogo di nascita, foto.

La scheda di un attore ha una lista di contenuti recenti al quale egli ha partecipato.

Frase Relative Ai Registri:

I registi sono identificati dal nome e cognome.

Dei registi rappresentiamo nome, cognome, età, data di nascita, luogo di nascita, foto.

La scheda di un regista ha una lista di contenuti recenti al quale egli ha partecipato

Frase Relative ai Cinema:

I cinema sono identificati dalla loro localizzazione (che include provincia, regione e indirizzo)

Dei cinema rappresentiamo il nome, localizzazione, contatti.

Ad ogni cinema è associata la composizione in sale.

Frase Relative alle Sale:

Le sale sono identificate dal numero e dal cinema di appartenenza.
Delle sale rappresentiamo il numero.

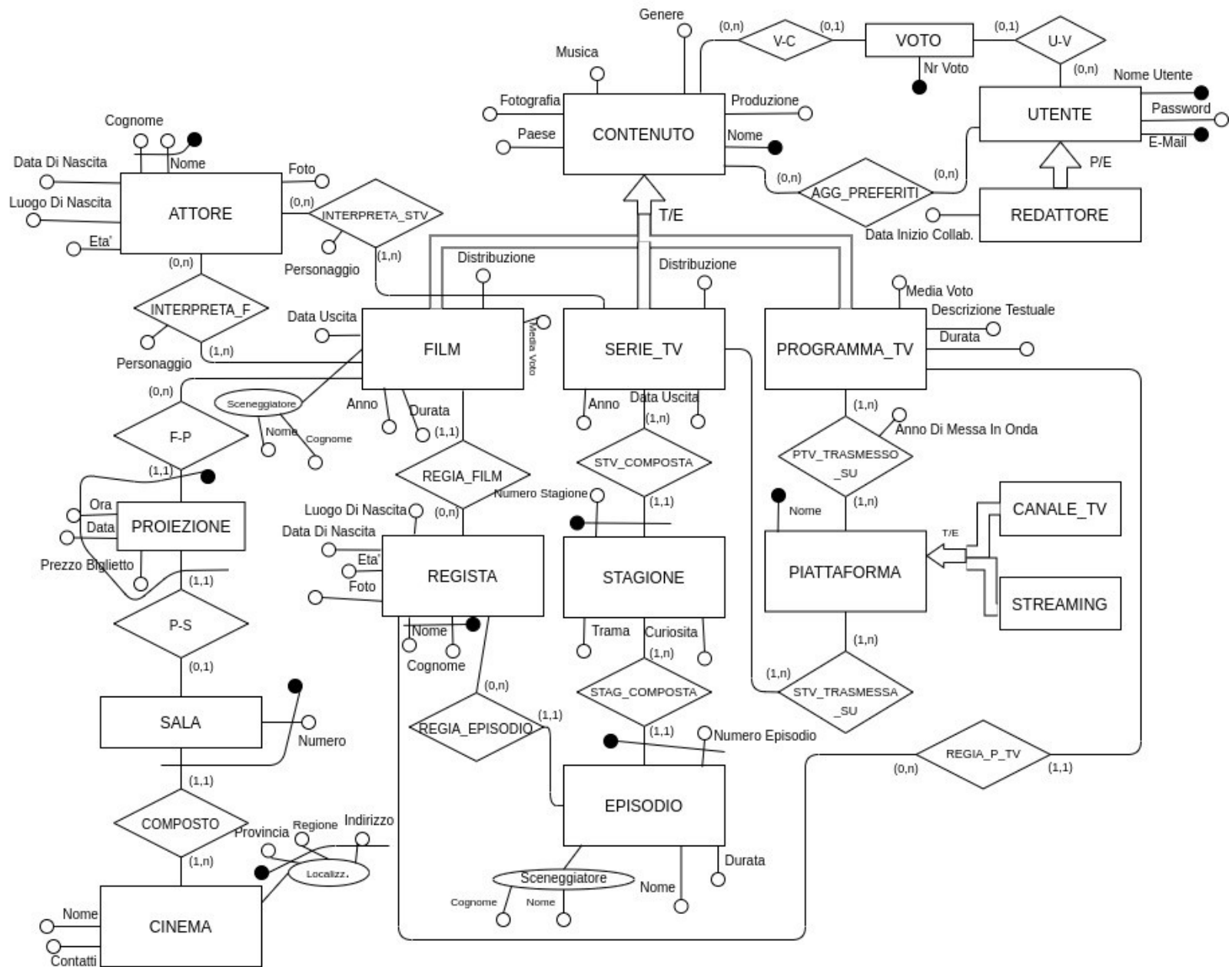
Frase Relative alle Piattaforme:

Le piattaforme sono identificate dal nome.
Le piattaforme sono divise in due tipi, canali TV e piattaforme di streaming video.

Operazioni sui Dati:

- Op. 1: Utente cerca un contenuto
- Op. 2: Utente visualizza un contenuto
- Op. 3: Utente visualizza lista di contenuti recenti al quale un attore ha partecipato
- Op. 4: Utente visualizza lista di contenuti recenti che un regista ha diretto
- Op. 5: Utente dà un voto al contenuto
- Op. 6: Utente inserisce un contenuto ai preferiti
- Op. 7: Utente rimuove un contenuto dai suoi contenuti preferiti
- Op. 8: Utente visualizza i suoi contenuti salvati
- Op. 9: Utente e suoi contenuti preferiti vengono cancellati dal database
- Op. 10: Redattore inserisce dati su un nuovo contenuto

1.4 Schema E-R e Business Rules



Vincoli di Integrita':

- 1.Regista non puo' avere cardinalita' minima 0 sia in REGIA_EPISODIO che in REGIA_PTV che in REGIA_FILM.
- 2.Attore non puo' avere cardinalita' minima 0 sia in INTERPRETA_F che in INTERPRETA_STV
- 3.Data Uscita di un film deve essere anteriore a Data in PROIEZIONE
- 4.Nr.Voto deve avere valori compresi tra 1 e 5

Vincoli di Derivazione:

- 1. Numero in SALA si ottiene contando le occorrenze di SALA che partecipano alla associazione COMPOSTO con CINEMA.
 - 2. Numero Stagione si ottiene contando le occorrenze di STAGIONE che partecipano alla associazione STV_COMPOSTA con SERIE_TV
 - 3. Numero Episodio si ottiene contando le occorrenze di EPISODIO che partecipano alla associazione STAG_COMPOSTA con STAGIONE
 - 4. Media Voto sia in PROGRAMMA_TV che in FILM si ottiene sommando I Nr. Voto dell entita' VOTO e dividendo per il numero di occorrenze di VOTO per quel contenuto (PROGRAMMA_TV O FILM)
-

2.1 Tavola Dei Volumi

CONCETTO	TIPO	VOLUME
Utente	E	10.000.000
Redattore	E	300
Voto	E	35.000.000
Contenuto	E	450.000
Programma TV	E	10.000
Piattaforma	E	50
Canale TV	E	30
Streaming	E	20
Serie TV	E	40.000
Stagione	E	120.000
Episodio	E	360.000
Film	E	400.000
Regista	E	380.000
Attore	E	4.500.000
Proiezione	E	24.000.000
Sala	E	4500
Cinema	E	1500
Composto	A	4500
P-S	A	24.000.000
F-P	A	24.000.000
Interpreta_F	A	4.000.000

Interpreta_STV	A	600.000
V-C	A	35.000.000
U-V	A	35.000.000
Agg_preferiti	A	3.600.000
Regia_Film	A	400.000
STV_Composta	A	120.000
PTV_Trasmesso_Su	A	20.000
Regia_episodio	A	360.000
Stag_composta	A	360.000
STV_trasmessa_su	A	120.000
Regia_ptv	A	10.000

Motivazioni:

- Pubblico italiano
- In media ogni utente da 3.5 voti
- Contenuti = 450.000 di cui 400.000 film, 40.000 serie TV e 10.000 programmi TV
- Piattaforme = 50 di cui 30 canali TV e 20 siti streaming
- Ogni Serie tv ha in media 3 stagioni, $40.000 \times 3 = 120.000$ stagioni
- Ogni stagione ha in media 13 episodi, $120.000 \times 13 = 360.000$ episodi
- Ogni film ha in media 10 attori, ogni serie tv ha in media 15 attori: $400.000 \times 10 + 40.000 \times 15 = 4.600.000$ attori arrotondiamo a 4.500.000 perche un attore puo' far parte di piu film/serie tv.
- I cinema in italia sono 1500, in media hanno 3 sale ciascuno, quindi le sale sono 4500
- le proiezioni avvengono solo per I film piu recenti, che supponiamo essere un 8% dei film , cioe' 32000, i piu attesi verranno proiettati in tutti I cinema, I meno attesi in pochi cinema quindi generalmente ho deciso che I film recenti vengono proiettati in 50% dei cinema (le proiezioni di un film possono essere tante, percio' le proiezioni sono cosi' alte) : $32000 \times 750 = 24.000.000$
- U-V e V-C avranno lo stesso volume di voto.
- Agg_preferiti ha cardinalita' 3.600.000 perche' in media ogni utente aggiunge ai preferiti 8 contenuti
- PTV_trasmesso_su ha cardinalita' 10.000 perche solitamente un programma tv viene trasmesso su un solo canale tv, quindi $10.000 + 1$ dei canali streaming in cui i PTV viene trasmesso = $10.000 + 1 \times 10.000 = 20.000$
- STV_trasmessa_su ha cardinalita' 120.000 perche viene trasmessa in 1 canale tv (40.000) + 40.000×2 (circa) piattaforme streaming
- Interpreta_F ha cardinalita 4.000.000 perche' abbiamo supposto che ogni film ha in media 10 attori, quindi 10×400.000
- Interpreta_STV ha cardinalita 600.000 perche' abbiamo supposto che ogni STV ha in media 15 attori, quindi 15×40.000

2.2 Tavola delle operazioni:

OPERAZIONE	DESCRIZIONE	TIPO	FREQUENZA
1	Utente cerca un contenuto	Interattiva	1.000.000
2	Utente visualizza un contenuto	Interattiva	100.000
3	tente visualizza lista di contenuti recenti al quale un attore ha partecipato	Interattiva	10.000
4	Utente visualizza lista di contenuti recenti che un regista ha diretto	Interattiva	10.000
5	Utente da un voto al contenuto	Interattiva	50.000
6	Utente inserisce un contenuto ai preferiti	Interattiva	50.000
7	Utente rimuove un contenuto dai suoi contenuti preferiti	Interattiva	30.000
8	Utente visualizza i suoi contenuti salvati	Interattiva	500.000
9	Utente e suoi contenuti preferiti vengono cancellati dal database	Interattiva	5.000
10	Redattore inserisce dati su un nuovo contenuto	Interattiva	1.000

2.3 Ristrutturazione dello schema E-R

-) 2.3.1 Analisi Delle Ridondanze

Ridondanze trovate:

- 1.L'attributo Numero dell'entita' Sala e' derivabile dall'associazione COMPOSIZIONE
- 2.L'attributo Numero Stagione dell'entita' Stagione e' derivabile dall'associazione STV_COMPOSTA
- 3.L'attributo Numero Episodio dell'entita' Episodio e' derivabile dall'associazione STAG_COMPOSTA
- 4.L'attributo Media Voto dell'entita Film e Programma e' derivabile dall'associazione V-C

Analisi della Ridondanza Media Voto:

(operazione 5: Utente da un voto al contenuto)

(operazione 2: Utente visualizza un contenuto)

SCENARIO A: (Assenza Della Ridondanza)

OPERAZIONE 5
SCHEMA DI OPERAZIONE

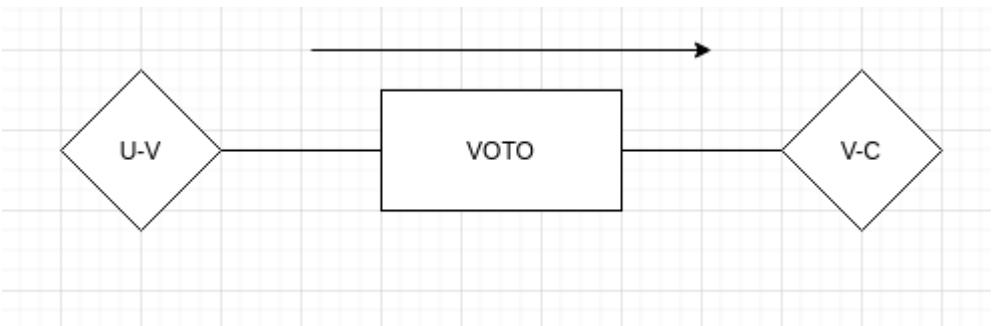


TAVOLA DEGLI ACCESSI:

CONCETTO	COSTRUTTO	ACCESSI	TIPO
U-V	Associazione	1	S
Voto	Entita'	1	S
V-C	Associazione	1	S

OPERAZIONE 2

SCHEMA DI OPERAZIONE

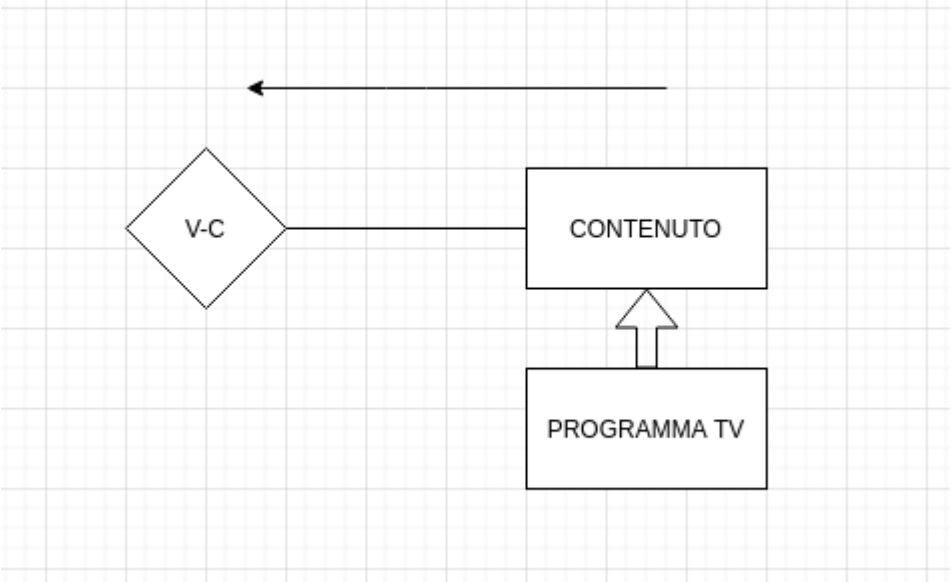


TAVOLA DEGLI ACCESSI

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Programma TV	Entita'	1	L
Contenuto	Entita	1	L
V-C	Associazione	78	L

78 accessi dell'associazione V-C perche' in media ogni contenuto ha $35.000.000/450.000 \approx 78$ Voti dati

SCENARIO B (PRESENZA DELLA RIDONDANZA):

OPERAZIONE 5
SCHEMA DI OPERAZIONE

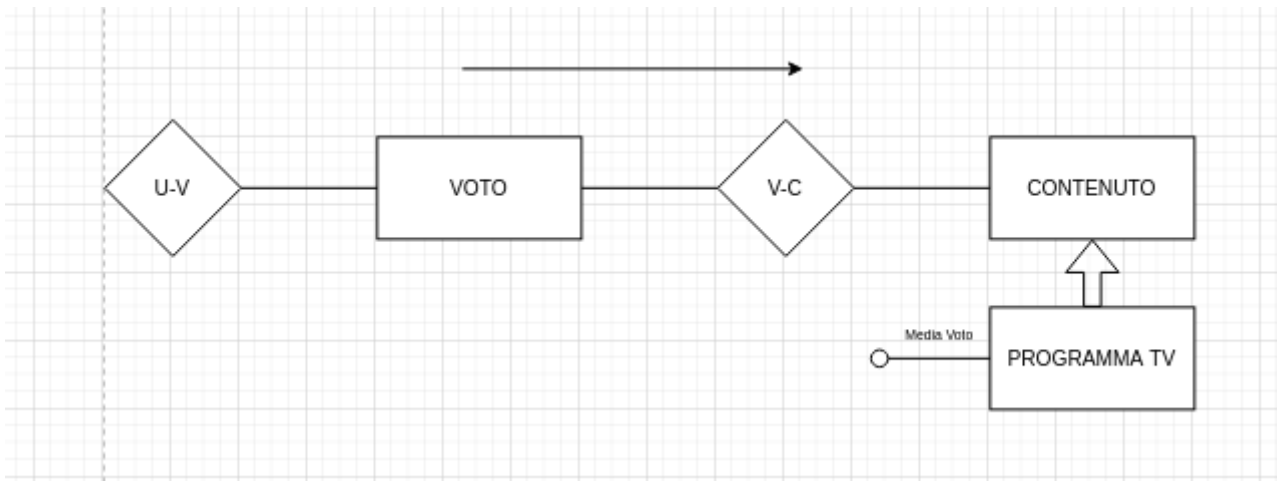


TAVOLA DEGLI ACCESSI

CONCETTO	COSTRUTTO	ACCESSI	TIPO
U-V	Associazione	1	S
Voto	Entita'	1	S
V-C	Associazione	1	S
Contenuto	Entita'	1	L
Programma TV	Entita'	1	L
Programma TV	Entita'	1	S

OPERAZIONE 2

SCHEMA DI OPERAZIONE

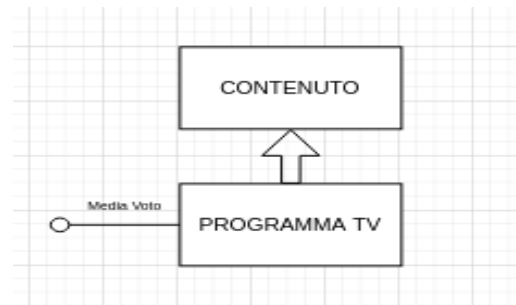


TAVOLA DEGLI ACCESSI

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Contenuto	Entita'	1	L
Programma TV	Entita'	1	L

Scenario A: Assenza Di Ridondanza:

- Spazio: 0 Byte
- Tempo:
 - Operazione 5: 3 (accessi) * 50.000 (operazioni al giorno) **accessi in scrittura al giorno**
 - Operazione 2: 78 (voti) * 100.000 (accessi) **accessi in lettura al giorno**

Contando gli accessi in scrittura doppi:

8.100.000 Accessi al giorno

Scenario B: Presenza Di Ridondanza:

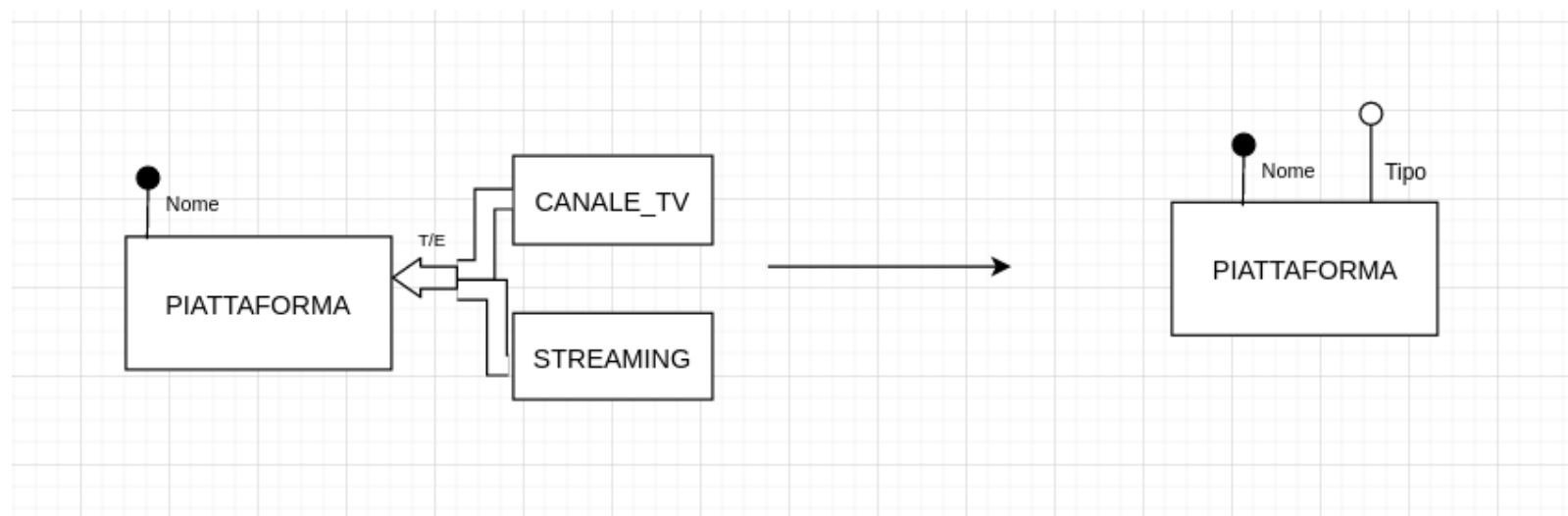
- Spazio: Assumendo di usare 4 Byte per memorizzare la Media Voto di ogni Programma TV:
4 * 10.000 (programmi TV) 400.000 Bytes
- Tempo:
 - Operazione 5: 4 accessi in S * 50.000 (accessi) + 2 accessi in L * 50.000 (accessi)
 - Operazione 2: 2 accessi in L * 100.000 (accessi)

Contando gli accessi in scrittura doppi:

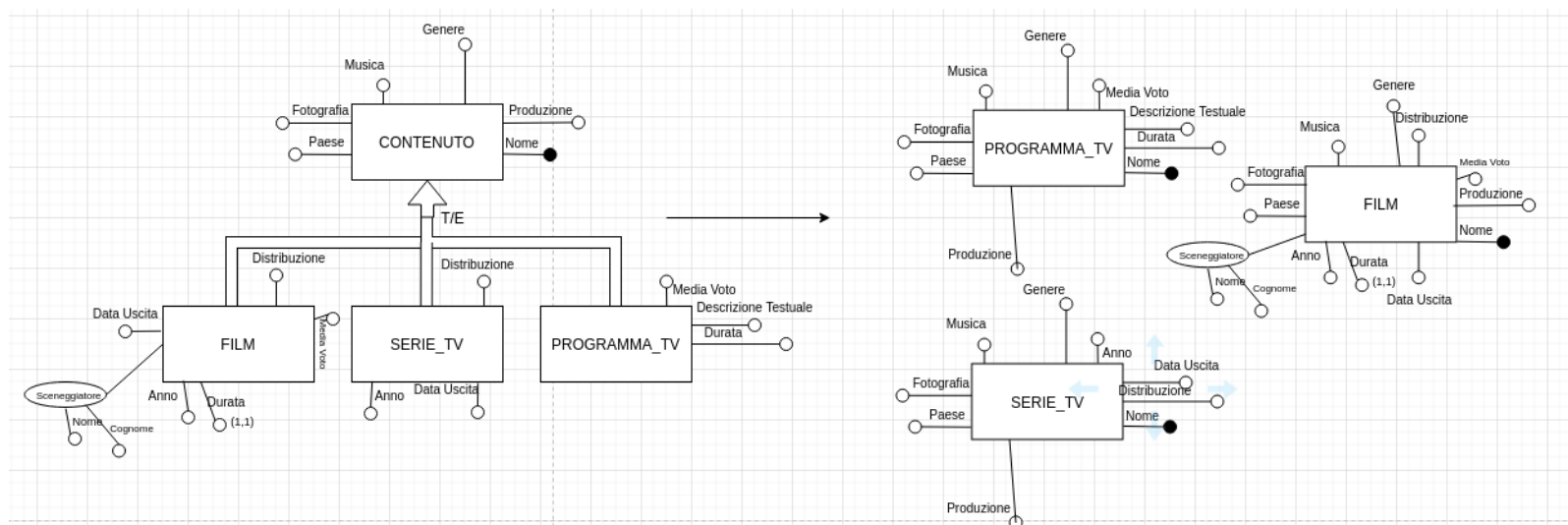
700.000 Accessi al giorno

Eliminando la ridondanza risparmierebbero meno di metà' di 1 Megabyte ma avremmo circa 7.400.000 accessi al giorno in piu' al database, quindi possiamo concludere che mantenere aggiornato il dato ridondante costa meno che ricalcolarlo.

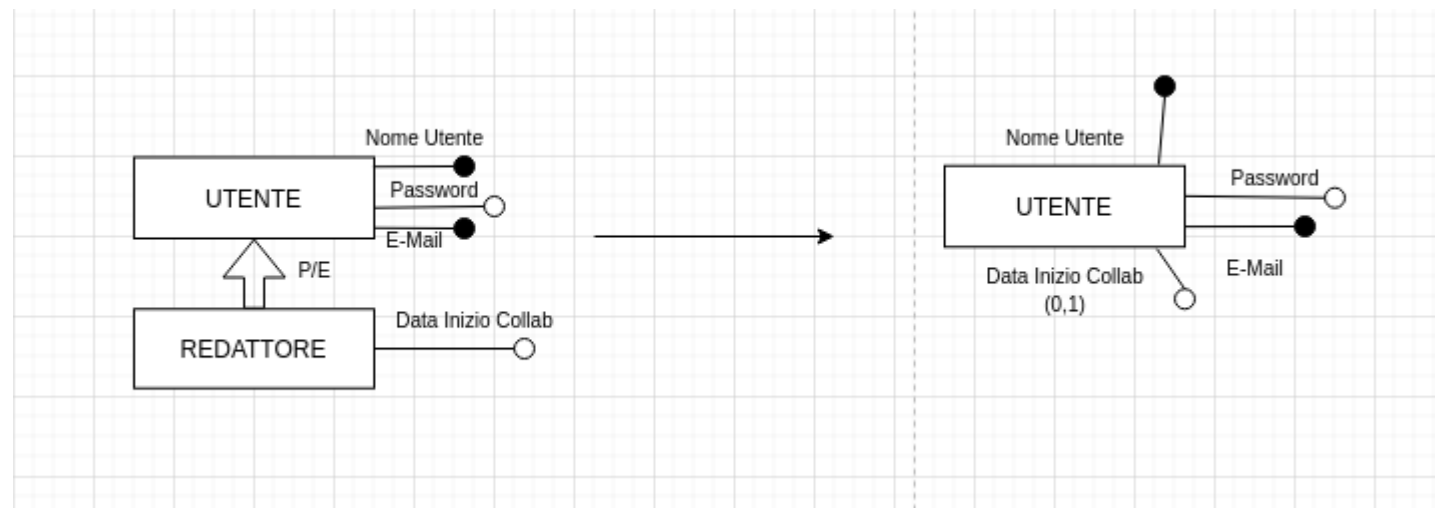
-) 2.3.2 Eliminazione Delle Generalizzazioni (con motivazione delle scelte effettuate)



Le operazioni non fanno distinzioni tra le entita' figlie, quindi accorpamo i figli nei genitori, e a Piattaforma viene aggiunto l'attributo Tipo che puo' assumere i valori di "Canale Tv" e "Streaming"



Ogni entita' figlia ha associazioni diverse nello schema ER, il che significa che gli accessi sulle entita' figlie sono distinti, perciò conviene accorpare il genitore nelle entita' figlie.

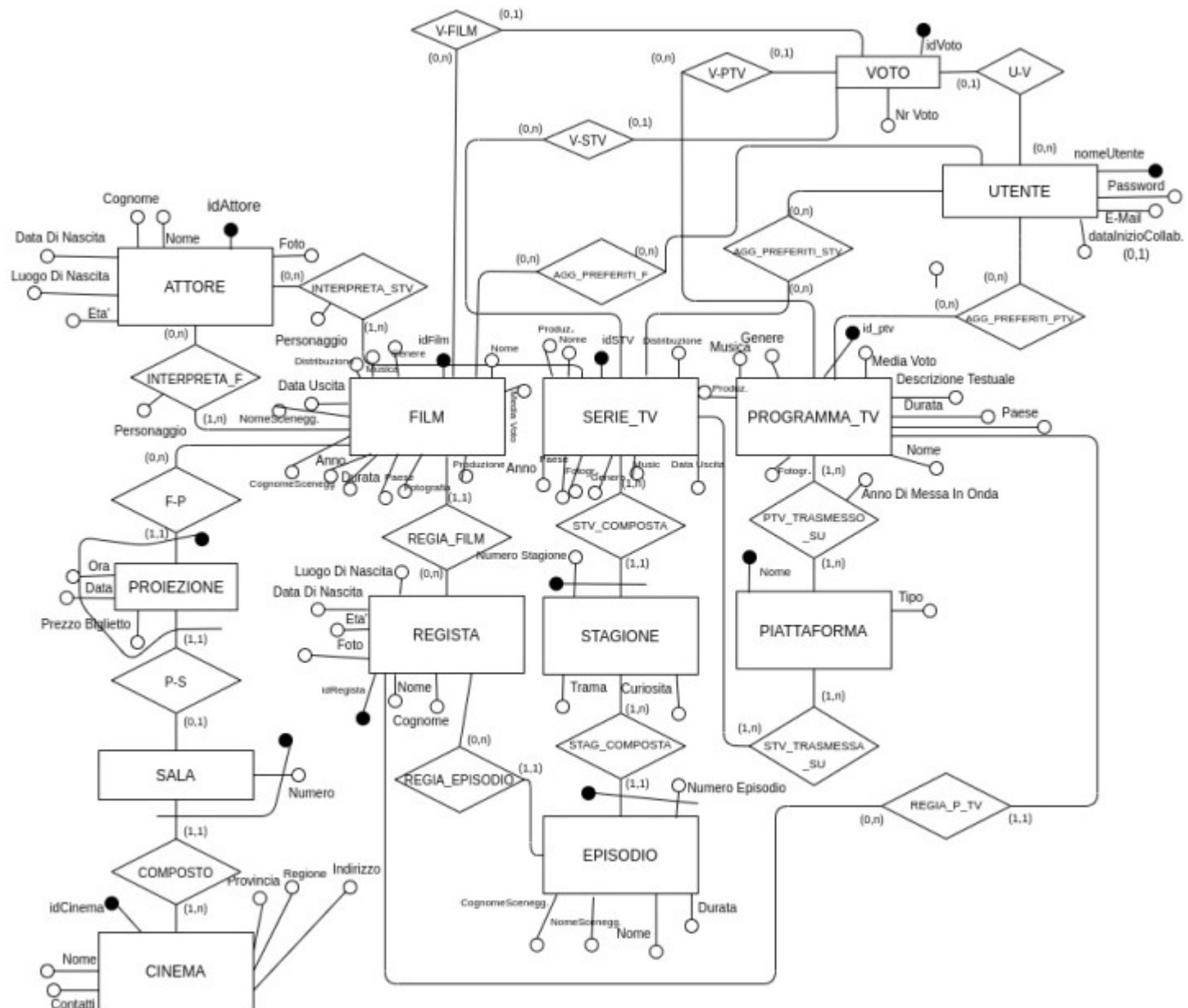


Le operazioni non fanno distinzione tra i figli, quindi decido di accorpare il figlio nel genitore, esso eredita l'attributo "Data Inizio Collab" che essendo opzionale assume valori (0,1), quindi o c'è o non c'è.

-) 2.3.3 Eventuale scelta degli identificatori principali (con motivazione delle scelte effettuate)

1. L'entità Utente ha come identificatore nomeUtente
2. Per l'entità Voto creo un identificatore surrogato per definire in modo univoco occorrenze di voti con valori uguali.
3. Per l'entità Film creo un identificatore surrogato perché potrebbero esserci Film con lo stesso nome
4. Per l'entità Film creo 2 attributi nomeSceneggiatore e cognomeSceneggiatore per eliminare l'attributo composto
4. Per l'entità Programma TV creo un identificatore surrogato perché potrebbero esserci Programma TV con lo stesso nome
5. Per l'entità Serie TV creo un identificatore surrogato perché potrebbero esserci Serie TV con lo stesso nome
6. Per l'entità Attore creo un identificatore surrogato perché potrebbero esserci attori con lo stesso nome
7. Per l'entità Regista creo un identificatore surrogato perché potrebbero esserci registi con lo stesso nome
8. sceneggiatore
9. Per l'entità cinema creo un identificatore surrogato coinvolgendo nell'identificazione non più i 3 attributi (Regione, provincia, indirizzo) ma 1 solo.
4. Per l'entità Episodio creo 2 attributi nomeSceneggiatore e cognomeSceneggiatore per eliminare l'attributo composto

-) 2.3.4 Schema E-R ristrutturato e Business Rules



Vincoli di Integrita':

- 1.Regista non puo' avere cardinalita' minima 0 sia in REGIA_EPISODIO che in REGIA_PTV che in REGIA_FILM.
- 2.Attore non puo' avere cardinalita' minima 0 sia in INTERPRETA_F che in INTERPRETA_STV
- 3.DataUscita di un film deve essere anteriore a Data in PROIEZIONE
- 4.Nr.Voto deve avere valori compresi tra 1 e 5

5. Solo gli utenti che hanno "Data Inizio Collab" a 1 possono modificare I dati di STV o Film
6. Per l'attributo Tipo in Piattaforma I valori possono essere o Canale Tv oppure Streaming.
7. Se Utente ha AGG_PREFERITI_F o AGG_PREFERITI_STV o AGG_PREFERITI_PTV tutti con cardinalita' 0, allora anche STV PTV e FILM avranno le loro cardinalita' delle rispettive associazioni a 0.

Vincoli Di Derivazione:

1. Numero in SALA si ottiene contando le occorrenze di SALA che partecipano alla associazione COMPOSTO con CINEMA.
2. Numero Stagione si ottiene contando le occorrenze di STAGIONE che partecipano alla associazione STV_COMPOSTA con SERIE_TV
3. Numero Episodio si ottiene contando le occorrenze di EPISODIO che partecipano alla associazione STAG_COMPOSTA con STAGIONE
4. Media Voto sia in PROGRAMMA_TV che in FILM si ottiene sommando I Nr. Voto dell entita' VOTO e dividendo per il numero di occorrenze di VOTO per quel contenuto (PROGRAMMA_TV O FILM)

-) 2.4 Schema Relazionale (Con Indicazione dei vincoli di integrita' referenziale)

Le Entita' diventano le seguenti relazioni:

UTENTE(NomeUtente, Password, E-Mail, DataInizioCollaborazione*)

VOTO(IdVoto, NrVoto, Utente*, PTV*, STV*, FILM*)

VOTO(Utente*) referencia Utente(NomeUtente)
 VOTO(PTV*) referencia PROGRAMMA_TV(IdPTV)
 VOTO(STV*) referencia SERIE_TV(IdSTV)
 VOTO(FILM*) referencia FILM(IdFILM)

FILM(IdFilm, Nome, MediaVoto*, Produzione, Fotografia, Paese, Durata, Anno, NomeSceneggiatore, CognomeSceneggiatore, DataUscita, Distribuzione, Musica, Genere, Regista)

FILM(Regista) referencia REGISTA(IdRegista)

SERIE_TV(IdSTV, Nome, Produzione, Paese, Anno, Fotografia, Genere, Musica, DataUscita, Distribuzione)

STAGIONE(NumeroStagione, STV Trama, Curiosita)

STAGIONE(STV) referencia STV(IdSTV)

EPISODIO(NumeroEpisodio, Stagione, STV, Nome, Durata, NomeSceneggiatore, CognomeSceneggiatore, Regista)

EPISODIO(Regista) referencia REGISTA(IdRegista)
 EPISODIO(Stagione) referencia STAGIONE(NumeroStagione)
 EPISODIO(STV) referencia SERIE_TV(IdSTV)

PROGRAMMA_TV(Id_PTV, Nome, MediaVoto*, DescrizioneTestuale, Durata, Paese, Fotografia, Produzione, Musica, Genere, Regista)
PROGRAMMA_TV(Regista) referencia REGISTA(IdRegista)

PIATTAFORMA(Nome, Tipo)

REGISTA(IdRegista, Nome, Cognome, Eta, DataDiNascita, LuogoDiNascita)

ATTORE(IdAttore, Nome, Cognome, DataDiNascita, LuogoDiNascita, Eta)

PROIEZIONE(Data, Ora, Film, Sala, PrezzoBiglietto)
PROIEZIONE(Film) referencia FILM(IdFilm)
PROIEZIONE(Sala) referencia SALA(Numero)

SALA(Numero, Cinema)
SALA(Cinema) referencia CINEMA(IdCinema)

CINEMA(IdCinema, Nome, Contatti, Provincia, Regione, Indirizzo)

Associazioni diventano le seguenti relazioni:

AGG_PREFERITI_F (<u>IdFilm</u> , <u>NomeUtente</u>)	AGG_PREFERITI_F(idFilm) referencia FILM(idFilm) AGG_PREFERITI_F(NomeUtente) referencia UTENTE(NomeUtente)
AGG_PREFERITI_STV (<u>idSTV</u> , <u>NomeUtente</u>)	AGG_PREFERITI_STV(idSTV) referencia SERIE_TV(idSTV) AGG_PREFERITI_STV(NomeUtente) referencia UTENTE(NomeUtente)
AGG_PREFERITI_PTV (<u>idPTV</u> , <u>NomeUtente</u>)	AGG_PREFERITI_PTV(idPTV) referencia PROGRAMMA_TV(idPTV) AGG_PREFERITI_PTV(NomeUtente) referencia UTENTE(NomeUtente)
PTV_TRASMESSO_SU (<u>Nome</u> , <u>idPTV</u> , AnnoDiMessalnOnda)	PTV_TRASMESSO_SU(Nome) referencia PIATTAFORMA(Nome) PTV_TRASMESSO_SU(idPTV) referencia PROGRAMMA_TV(idPTV)
STV_TRASMESSA_SU (<u>Nome</u> , <u>idSTV</u>)	STV_TRASMESSA_SU(Nome) referencia PIATTAFORMA(Nome) STV_TRASMESSA_SU(idSTV) referencia SERIE_TV(idSTV)
INTERPRETA_STV (<u>IdSTV</u> , <u>IdAttore</u> , Personaggio)	INTERPRETA_STV(idSTV) referencia SERIE_TV(idSTV) INTERPRETA_STV(idAttore) referencia ATTORE(IdAttore)
INTERPRETA_F (<u>IdFilm</u> , <u>IdAttore</u> , Personaggio)	INTERPRETA_F(idFilm) referencia FILM(idFilm) INTERPRETA_F(idAttore) referencia ATTORE(IdAttore)

-)3. Implementazione

3.1) DDL di Creazione del Database

```
create domain intPassword as varchar(30) not null check (value >= '%%%%%%%%%');
create domain email as varchar(30) not null check (value>='%%%%%%%%%');
create table UTENTE(
    NomeUtente varchar(20),
    Password intPassword,
    EMail email,
    DataInizioCollaborazione date,
    constraint primary_key_utente primary key(NomeUtente)
);
```

```
create sequence SequenzaSTV
increment by 1 start 1;
create domain STV_id as integer default nextval('SequenzaSTV');
create table SERIE_TV(
    IdSTV STV_id,
    Nome varchar(25),
    Produzione varchar(25) not null,
    Paese varchar(25) not null,
    Anno integer not null,
    Fotografia varchar(25) not null,
    Genere varchar(25) not null,
    Musica varchar(25) not null,
    DataUscita date not null,
    Distribuzione varchar(25) not null,
    constraint primary_key_STV primary key(IdSTV)
);
```

```
create table STAGIONE(
    NumeroStagione numeric(2),
    STV STV_id,
    Trama varchar(500) not null,
    Curiosita varchar(250) not null,
    constraint primary_key_Stagione primary key (STV, NumeroStagione),
    constraint stagione_foreign_key_STV foreign key(STV) references SERIE_TV(IdSTV)
    on delete cascade
);
```

```
create sequence SequenzaREGISTA
increment by 1 start 1;
create domain REGISTA_id as integer default nextval('SequenzaREGISTA');
create table REGISTA(
    IdRegista REGISTA_id primary key,
    Nome varchar(25) not null,
    Cognome varchar(25) not null,
    Eta decimal(2) not null,
```

```
        DataDiNascita date not null,  
        LuogoDiNascita varchar(25) not null  
    );
```

```
create sequence SequenzaPTV  
increment by 1 start 1;  
create domain PTV_id as integer default nextval('SequenzaPTV');  
create table PROGRAMMA_TV(  
    IdPTV PTV_id primary key,  
    Nome varchar(25) not null,  
    MediaVoto numeric(3,2),  
    DescrizioneTestuale varchar(200) not null,  
    Durata interval hour to minute not null,  
    Paese varchar(25) not null,  
    Fotografia varchar(25) not null,  
    Produzione varchar(25) not null,  
    Musica varchar(25) not null,  
    Genere varchar(25) not null,  
    Regista REGISTA_id,  
    constraint PTV_foreign_key_REGISTA foreign key(Regista) references REGISTA(IdRegista)  
);
```

```
create sequence SequenzaFILM  
increment by 1 start 1;  
create domain Film_id as integer default nextval('SequenzaFILM');  
create table film(  
    IdFilm Film_id primary key,  
    Nome varchar(30) not null,  
    MediaVoto numeric (3,2),  
    Produzione varchar(30) not null,  
    Fotografia varchar(30) not null,  
    Paese varchar(30) not null,  
    Durata interval to hour to minute not null,  
    Anno integer not null,  
    NomeSceneggiatore varchar(30) not null,  
    CognomeSceneggiatore varchar(30) not null,  
    DataUscita date not null,  
    Distribuzione varchar(30) not null,  
    Musica varchar(30) not null,  
    Genere varchar(30) not null,  
    Regista REGISTA_id,  
    constraint FILM_foreign_key_REGISTA foreign key(Regista) references REGISTA(IdRegista)  
    on delete no action  
);
```

```

create table EPISODIO(
    NumeroEpisodio numeric(2),
    Stagione numeric(2),
    STV varchar(25),
    Nome varchar(25) not null,
    Durata interval to hour to minute not null,
    NomeSceneggiatore varchar(25) not null,
    CognomeSceneggiatore varchar(25) not null,
    Regista REGISTA_id,
    constraint episodio_primary_key primary key(STV,Stagione,NumeroEpisodio),
    constraint episodio_foreign_key_REGISTA foreign key(Regista) references REGISTA(IdRegista)
    on delete no action
    constraint episodio_foreign_key_STAGIONE foreign key (STV) references STAGIONE(STV),
    constraint episodio_foreign_key_STAGIONE foreign key (Stagione) references STAGIONE(NumeroStagione)
    on delete cascade
);

```

```

create sequence SequenzaAttori
increment by 1 start 1;
create domain ATTORI_id as integer default nextval('SequenzaAttori');
create table ATTORE(
    IdAttore ATTORI_id primary key,
    Nome varchar(30) not null,
    Cognome varchar(30) not null,
    DataDiNascita date not null,
    LuogoDiNascita varchar(30) not null,
    Eta decimal (2) not null
);

```

```

create sequence SequenzaVoti
increment by 1 start 1;
create domain VOTO_id as integer default nextval('SequenzaVoti');
create domain val_voto as integer not null check (value>'0' and value <=5);
create table VOTO(
    IdVoto VOTO_id primary key,
    NrVoto val_voto,
    Utente varchar(20),
    PTV PTV_id default null,
    STV STV_id default null,
    FILM FILM_id default null,
    constraint voto_foreign_key(Utente) references UTENTE(NomeUtente)
    on delete set null
    on update cascade
    constraint voto_foreign_key(STV) references SERIE_TV(IdSTV)
    on delete cascade
    on update cascade
    constraint voto_foreign_key(PTV) references PROGRAMMA_TV(IdPTV)
    on delete cascade
    on update cascade
    constraint voto_foreign_key(FILM) references FILM(IdFILM)
    on delete cascade
);

```

```
on update cascade,  
constraint check_not_multiple check ((PTV is null and STV is null and FILM is not null) or  
                                     (PTV is null and STV is not null and FILM is null) or  
                                     (PTV is not null and STV is null and FILM is null) )  
);
```

```
create sequence SequenzaCinema  
increment by 1 start 1;  
create domain CINEMA_id as integer default nextval('SequenzaCinema');  
create table CINEMA(  
    IdCinema CINEMA_id primary key,  
    Nome varchar(30) not null,  
    Contatti varchar(30) not null,  
    Provincia varchar(30) not null,  
    Regione varchar(30) not null,  
    Indirizzo varchar(30) not null  
);
```

```
create table SALA(  
    Numero decimal (2),  
    Cinema CINEMA_id,  
    constraint sala_primary_key primary key(Numero, Cinema),  
    constraint sala_foreign_key_Cinema foreign key(Cinema) references CINEMA(IdCinema)  
    on delete cascade  
);
```

```
create table PROIEZIONE(  
    Datapr date,  
    Ora time(2),  
    Film FILM_id,  
    Sala decimal (2),  
    Cinema CINEMA_id,  
    PrezzoBiglietto numeric(4,2) not null,  
    constraint proiezione_primary_key primary key(Datapr, Ora, Film, Sala, Cinema),  
    constraint proiezione_foreign_key_Film foreign key(Film), references FILM(IdFilm)  
    on delete cascade  
    on update cascade  
    constraint proiezione_foreign_key_sala foreign key(Sala), references SALA(Numero),  
    constraint proiezione_foreign_key_Cinema foreign key(Cinema), references CINEMA(IdCinema)  
    on delete cascade  
    on update cascade  
);
```

```
create table PIATTAFORMA(  
    Nome varchar (20) primary key,  
    Tipo varchar(9) not null,  
    constraint check_tipo check (Tipo like 'Canale TV' or Tipo like 'Streaming')  
);
```

```
create table INTERPRETA_STV(  
    IdSTV STV_id,  
    IdAttore ATTORI_id,  
    Personaggio varchar(20) not null,  
    constraint interpreta_stv_primary_key primary key(IdSTV, IdAttore),  
    constraint interpreta_stv_foreign_key_stv foreign key (IdSTV) references SERIE_TV(IdSTV)  
    on delete cascade  
    constraint interpreta_stv_foreign_key_attore foreign key(IdAttore) references ATTORE(IdAttore)  
    on delete cascade  
);
```

```
create table INTERPRETA_F(  
    IdFILM FILM_id,  
    IdAttore ATTORI_id,  
    Personaggio varchar(20) not null,  
    constraint interpreta_f_primary_key primary key(IdFILM, IdAttore),  
    constraint interpreta_f_foreign_key_f foreign key (IdFILM) references FILM(IdFilm)  
    on delete cascade  
    constraint interpreta_f_foreign_key_attore foreign key(IdAttore) references ATTORE(IdAttore)  
    on delete cascade  
);
```

```
create table PTV_TRASMESSO_SU(  
    NomePiatt varchar (9),  
    IdPTV PTV_id,  
    AnnoDiMessalnOnda numeric(4) not null,  
    constraint ptv_trasmesso_su_primary_key primary key(NomePiatt, IdPTV),  
    constraint ptv_trasmesso_su_foreign_key_PTV foreign key(IdPTV) references PROGRAMMA_TV(IdPTV)  
    on delete cascade  
    constraint ptv_trasmesso_su_foreign_key_piattaforma foreign key (NomePiatt) references PIATTAFORMA(Nome)  
    on delete cascade  
);
```

```
create table STV_TRASMESSO_SU(  
    Nome varchar (9),  
    IdSTV STV_id,  
    constraint stv_trasmessa_su_primary_key primary key(Nome, IdSTV),  
    constraint stv_trasmessa_su_foreign_key_piattaforma foreign key(Nome) references PIATTAFORMA(Nome)  
    on delete cascade  
    constraint stv_trasmessa_su_foreign_key_STV foreign key(IdSTV) references SERIE_TV(IdSTV)  
    on delete cascade  
);
```

```
create table AGG_PREFERITI_F(  
    IdFilm FILM_id;
```

```

NomeUtente varchar(20);
constraint agg_pref_f_primary_key primary key (NomeUtente, IdFilm);
constraint agg_pref_f_foreign_key_Utente foreign key(NomeUtente) references UTENTE(NomeUtente)
on delete cascade
constraint agg_pref_f_foreign_key_film foreign key(IdFilm) references FILM(IdFilm)
on delete cascade
);

create table AGG_PREFERITI_STV(
    IdSTV STV_id;
    NomeUtente varchar(20);
    constraint agg_pref_stv_primary_key primary key (NomeUtente, IdSTV);
    constraint agg_pref_stv_foreign_key_Utente foreign key(NomeUtente) references UTENTE(NomeUtente)
    on delete cascade
    constraint agg_pref_stv_foreign_key_stv foreign key(IdSTV) references SERIE_TV(IdSTV)
    on delete cascade
);

create table AGG_PREFERITI_PTV(
    IdPTV PTV_id;
    NomeUtente varchar(20);
    constraint agg_pref_ptv_primary_key primary key (NomeUtente, IdPTV);
    constraint agg_pref_ptv_foreign_key_Utente foreign key(NomeUtente) references UTENTE(NomeUtente)
    on delete cascade
    constraint agg_pref_stv_foreign_key_ptv foreign key(IdPTV) references PROGRAMMA_TV(IdPTV)
    on delete cascade
);

```

3.1) DML di popolamento di tutte le tabelle del database

```

insert into UTENTE(NomeUtente, Password, E-Mail, DataInizioCollaborazione*)
    values ('ledzepp', 'shaudhaudwa', 'ledzepp@gmail.com');

insert into UTENTE(NomeUtente, Password, E-Mail, DataInizioCollaborazione*)
    values ('eminiem22', '8877hh66vv', 'eminem22@gmail.com');

insert into UTENTE(NomeUtente, Password, E-Mail, DataInizioCollaborazione*)
    values ('allmetica87', 'pimpumpam69', 'allmetica87@hotmail.com');

insert into SERIE_TV(Nome, Produzione, Paese, Anno, Fotografia, Genere, Musica,
    DataUscita,Distribuzione) values
    ('breaking bad', 'xsysdhad', 'usa', '2008', 'martin black, thriller', 'van halen', '2008', 'centuryfox');

insert into STAGIONE(NumeroStagione, STV, Trama, Curiosita) values
    ('2', 'breaking bad', 'ssssssssssssssssssssssssssssssss', 'swwwwwwwwwwwwwwwwwwww');

insert into PROGRAMMA(Nome, Genere, Paese, Produzione, DescrizioneTestuale, Durata)
    values ('CiaoD', 'Comico', 'Italia', 'Mediaset', 'Descr1', '3:10');

```

```

insert into    REGISTA(Nome,Cognome,Età,LuogoNascita,DataNascita)
values ('Alex', 'Pena', '63','Madrid','1956-07-02')

insert into    FILM(Nome,DataUscita,Genere,Paese,Durata, Produzione,Muscihe,Fotografia,Distribuzione,Regista)
values ('Avengers:endgame','2018-04-22','Azione', 'USA', '3:01', 'Produzione1', 'Musiche1',
'Fotografo1','Distribuzione','2');

insert into    EPISODIO(SerieTv,Stagione,NumEpisodio, Titolo,Durata,NomeSceneggiatore,CognomeSceneggiatore,Regista)
values ('mr Robot', '01','01','hell0 w0rld', '0:51', 'mario', 'Kee', '1'

insert into    ATTORE(Nome,Cognome,Età,LuogoNascita,DataNascita)
values ('Rami', 'Malek', '30', 'cairo', '1990-02- 03');

insert into    VOTO(NumVoto,Utente,SerieTv)
values ('3', 'ariana grande', 'La Casa Di Carta');

insert into    VOTO(NumVoto,Utente,Programma)
values ('5','Brunovesta','1');

insert into    CINEMA(Nome,Contatti,Regione,Provincia,Indirizzo)
values ('cineplex torino', '088377493', 'Piemonte','Torino','Via Verdi');

insert into    SALA(Numero,Cinema)
values ('02','1');
insert into    SALA(Numero,Cinema)
values ('06','2');

insert into    PROIEZIONE(Data_p,Ora,Cinema,Sala,Film,Prezzo)
values ('2019-03-22','20:00', '1', '03', '6', '10.00');

insert into    PIATTAFORMA(Nome,Tipo)
values ('HULU', 'Streaming');

insert into    SCEN_F(Film,NomeSceneggiatore, CognomeSceneggiatore)
values ('1', 'John', 'Wick');

insert into    INTERPRETA_F(Attore,Film,Personaggio)
values ('9','6','bartender');

insert into    PREFERITI_SE(Utente,SerieTv)
values ('santo santi','rebelde');

```


3.1) Alcune Operazioni di cancellazione e modifica

1. cancellare un utente:

```
delete from Utente  
      where nomeUtente = 'santo santi'
```

2. Modifica film:

```
Update FILM  
set      Nome = 'Cosmopolitan: redemption'  
      Genere = 'Thriller'  
      Data Uscita = '23-01-2000'  
where IdFilm = '6'
```