

# **RiverFlix**

## **Piattaforma di Streaming di film e serie TV**

*Elisa Casalini*  
0001087239  
*elisa.casalini@studio.unibo.it*

*Mattia Lodi*  
0001020617  
*mattia.lodi3@studio.unibo.it*

*Francesco Testa*  
0001029406  
*francesco.testa3@studio.unibo.it*

<b>1. Analisi dei requisiti.....</b>	<b>3</b>
1.1. Requisiti espressi in linguaggio naturale.....	3
1.2. Glossario dei termini.....	3
1.3. Strutturazione dei requisiti.....	4
1.3.1. Frasi aspecifiche.....	4
1.3.2. Frasi relative ai prodotti.....	4
1.3.3. Frasi relative agli account.....	5
1.3.4. Frasi relative agli utenti.....	5
1.3.5. Frasi relative al personale.....	5
1.4. Specifica delle operazioni.....	5
<b>2. Progettazione concettuale.....</b>	<b>6</b>
2.3 Identificazioni delle entità e relazioni (top-down).....	6
2.4 Scheletro dello schema (bottom-up).....	6
2.5 Sviluppo componenti scheletro (inside-out).....	6
2.5.1 Utente.....	6
2.5.2 Prodotto cinematografico.....	7
2.5.3 Personale.....	8
2.6 Unione componenti.....	9
2.1. Dizionario dei dati.....	10
2.2. Regole aziendali.....	11
<b>3. Progettazione logica.....</b>	<b>12</b>
3.1. Tabella dei volumi.....	12
3.2. Tabella delle operazioni.....	13
3.3. Ristrutturazione dello schema concettuale.....	13
3.3.1. Rimozione delle ridondanze.....	13
3.3.2. Rimozione delle gerarchie.....	15
3.3.3. Accorpamento.....	16
3.3.4. Rimozione degli attributi multivalore.....	16
3.3.5. Identificazione delle chiavi primarie.....	17
Lo schema ER alla fine della ristrutturazione risulta essere:.....	17
3.4. Normalizzazione.....	18
3.5. Traduzione nel modello relazionale.....	18
<b>4. Codifica SQL.....</b>	<b>20</b>
4.1. Definizione dello schema.....	20
4.2. Codifica delle operazioni.....	23
<b>5. Testing.....</b>	<b>25</b>

# 1. Analisi dei requisiti

## 1.1. Requisiti espressi in linguaggio naturale

Si vuole realizzare una base di dati per un'azienda che fornisce un servizio di streaming di film e serie tv ai suoi abbonati.

In particolare si vuole tenere traccia di tutti i prodotti offerti, cioè film e serie TV, avendo per ognuno nome, codice, data di caricamento, categoria (CARA), budget, rating, location, genere, persone che hanno lavorato al film.

In caso di serie TV è importante dividere il prodotto in stagioni ed episodi, categorizzandoli singolarmente.

Per ogni account, avente id, email e piano di abbonamento, il servizio offre la possibilità di avere diversi utenti, ciascuno con un nome. Per ogni utente si tiene traccia dei prodotti visionati (in precedenza e attualmente in visione), e per ciascuno di questi, del watchtime e del gradimento (ottenuto dalle recensioni lasciate), si è anche interessati alla profilazione degli utenti: età, lingua, posizione, tempo di utilizzo.

Gli utenti possono inoltre aggiungere i prodotti che vorranno vedere alla watchlist e lasciare delle recensioni ad ogni prodotto.

Il personale può essere di tre tipi: produttori, registi o attori e per gli attori si vuole avere il ruolo interpretato nei film/serie tv nelle quali hanno recitato.

Per tutti si deve sapere il nome, la data di nascita, la nazionalità e lo stato di residenza.

## 1.2. Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
film	prodotto audiovisivo presente nell'offerta della piattaforma	prodotto	categoria, location, genere, personale
serie TV	prodotto audiovisivo presente nell'offerta della piattaforma, composto da stagioni ed episodi	prodotto	categoria, location, genere, personale, episodio, stagione
nome	titolo del prodotto	-	prodotto
classificazione	classificazione per il pubblico	CARA	prodotto
budget	costo di produzione del prodotto	-	prodotto
rating	voto IMDB o valutazione media degli utenti	-	prodotto

location	luoghi in cui è stato girato	-	prodotto
genere	tipo del prodotto	-	prodotto
personale	persone che hanno lavorato ad un prodotto	-	prodotto
episodio	parte di un prodotto seriale che viene gestito singolarmente	-	serie TV, stagione
stagione	insieme di episodi rilasciati a distanza ravvicinata	-	serie TV, episodio
account	identità digitale che può accogliere più persone fisiche	-	utente
utente	persona fisica parte di un account	-	prodotto
piano di abbonamento	tipologia del piano di abbonamento sottoscritto	-	account
watchtime	tempo di visione effettiva di un prodotto	-	utente, prodotto
gradimento	valutazione di un utente relativa ad un prodotto	-	utente, prodotto
watchlist	insieme dei prodotti che si desidera vedere in futuro	-	utente, prodotto
recensione	input dell'utente che ne esprime il gradimento	-	utente, prodotto
produttore	persona/entità che finanzia un prodotto	-	prodotto
regista	persona che dirige lo sviluppo di un prodotto	-	prodotto
attore	persona che recita in un prodotto	-	prodotto, ruolo
ruolo	parte di un attore in un prodotto	-	attore

### 1.3. Strutturazione dei requisiti

#### 1.3.1. Frasi aspecifiche

“Si vuole realizzare una base di dati per un’azienda che fornisce un servizio di streaming di film e serie tv ai suoi abbonati.”

#### 1.3.2. Frasi relative ai prodotti

“In particolare si vuole tenere traccia di tutti i prodotti offerti, cioè film e serie TV, avendo per ognuno nome, codice, data di caricamento, categoria (CARA), budget, rating, location, genere, persone che hanno lavorato al film.”

“In caso di serie TV è importante dividere il prodotto in stagioni ed episodi, categorizzandoli singolarmente.”

**1.3.3. Frasi relative agli account**

“Per ogni account, avente id, email e piano di abbonamento, il servizio offre la possibilità di avere diversi utenti, ciascuno con un nome.”

**1.3.4. Frasi relative agli utenti**

“Per ogni utente si tiene traccia dei prodotti visionati (in precedenza e attualmente in visione), e per ciascuno di questi, del watchtime e del gradimento (ottenuto dalle recensioni lasciate), si è anche interessati alla profilazione degli utenti: età, lingua, posizione, tempo di utilizzo.

Gli utenti possono inoltre aggiungere i prodotti che vorranno vedere alla watchlist e lasciare delle recensioni ad ogni prodotto.”

**1.3.5. Frasi relative al personale**

“Il personale può essere di tre tipi: produttori, registi o attori e per gli attori si vuole avere il ruolo interpretato nei film/serie tv nelle quali hanno recitato.

Per tutti si deve sapere il nome, la data di nascita, la nazionalità e lo stato di residenza.”

## **1.4. Specifica delle operazioni**

1. Aggiunta di un prodotto (4 volta a settimana)
2. Rimozione di un prodotto (2 volte a settimana)
3. Aggiornamento di un prodotto (2 volta a settimana)
4. Aggiornamento del rating (1 volta al giorno)
5. Aggiunta di un account (più volte al giorno)
6. Cambio abbonamento (1 volta al giorno)
7. Rimozione di un account (1 volta a settimana)
8. Aggiunta di un utente (6 volte al giorno)
9. Ricerca della top ten dei prodotti visionati (10 volte al giorno)
10. Ricerca della top ten dei prodotti visionati del mese (10 volte al giorno)
11. Ricerca prodotti consigliati (100 volte al giorno)
12. Ricerca di un prodotto (50 volte al giorno)
13. Visione di un prodotto (150 volte al giorno)
14. Valutazione di un prodotto (5 volte al giorno)
15. Informazioni di un prodotto (10 volte al giorno)

Le operazioni dipendono molto dal numero di utenti e di prodotti, che inizialmente stimiamo essere rispettivamente: 1000 e 5000

## 2. Progettazione concettuale

### 2.3 Identificazioni delle entità e relazioni (top-down)

Sono state identificate inizialmente le seguenti entità:

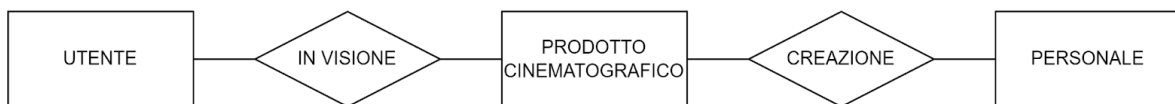
UTENTE, ACCOUNT, PRODOTTO CINEMATOGRAFICO, FILM, SERIE TV, EPISODI, PERSONALE, ATTORE, REGISTA, PRODUTTORE.

Di queste, 3 sono entità fondamentali:

- UTENTE, colui che fa uso della piattaforma streaming,
- PRODOTTO CINEMATOGRAFICO, che viene esposto all'interno della piattaforma,
- PERSONALE, i creatori del prodotto.

### 2.4 Scheletro dello schema (bottom-up)

Considerando solo le tre entità fondamentali ecco il primo scheletro di schema:

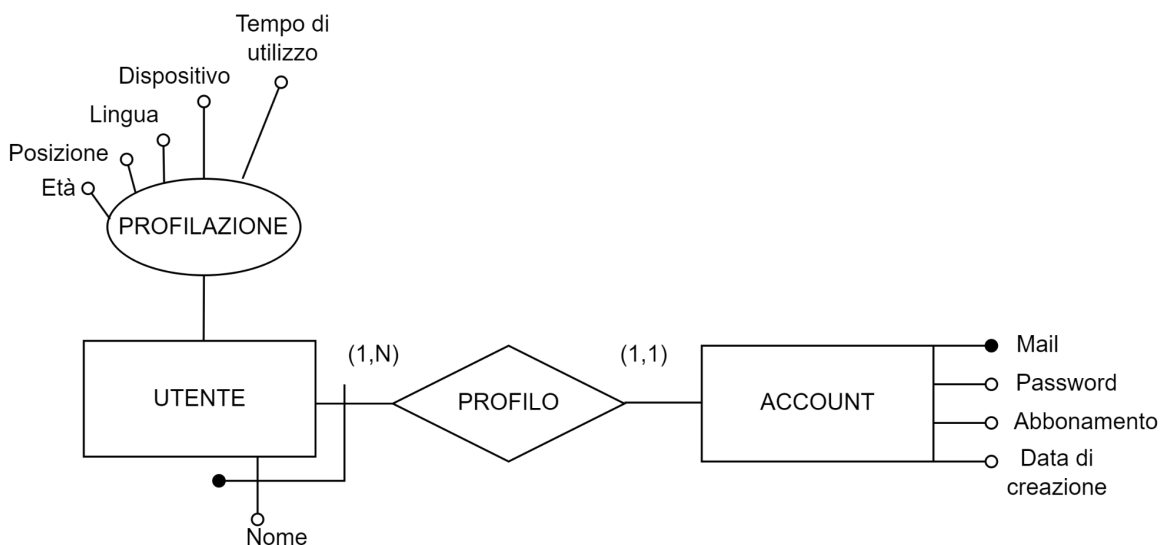


IN VISIONE è il prodotto cinematografico che sta guardando l'utente in questo momento,

CREAZIONE è l'insieme delle persone che si sono occupate della creazione del prodotto cinematografico.

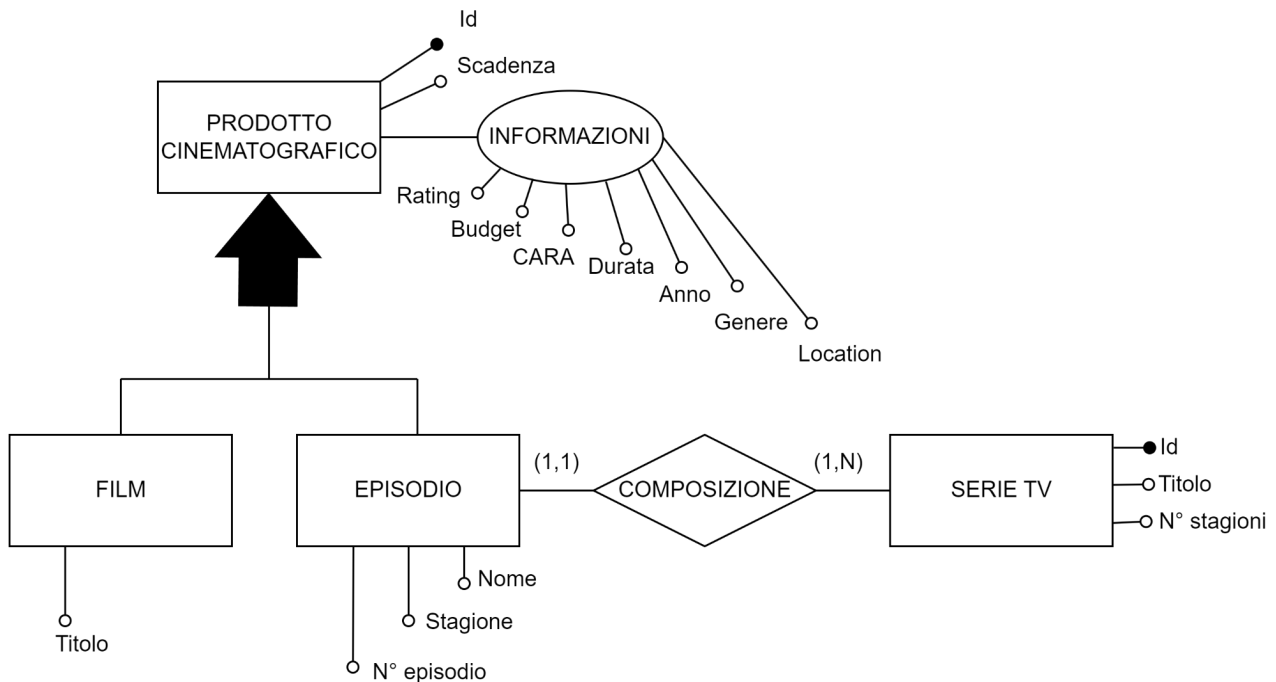
### 2.5 Sviluppo componenti scheletro (inside-out)

#### 2.5.1 Utente



PROFILO associa la persona all'account, da notare come per un account ci possano essere più utenti associati, ma un utente può avere un solo account. Abbiamo scelto di creare un attributo composto per tutti i dati di profilazione dell'utente in quanto molto preziosi per la nostra casa cinematografica.

## 2.5.2 Prodotto cinematografico

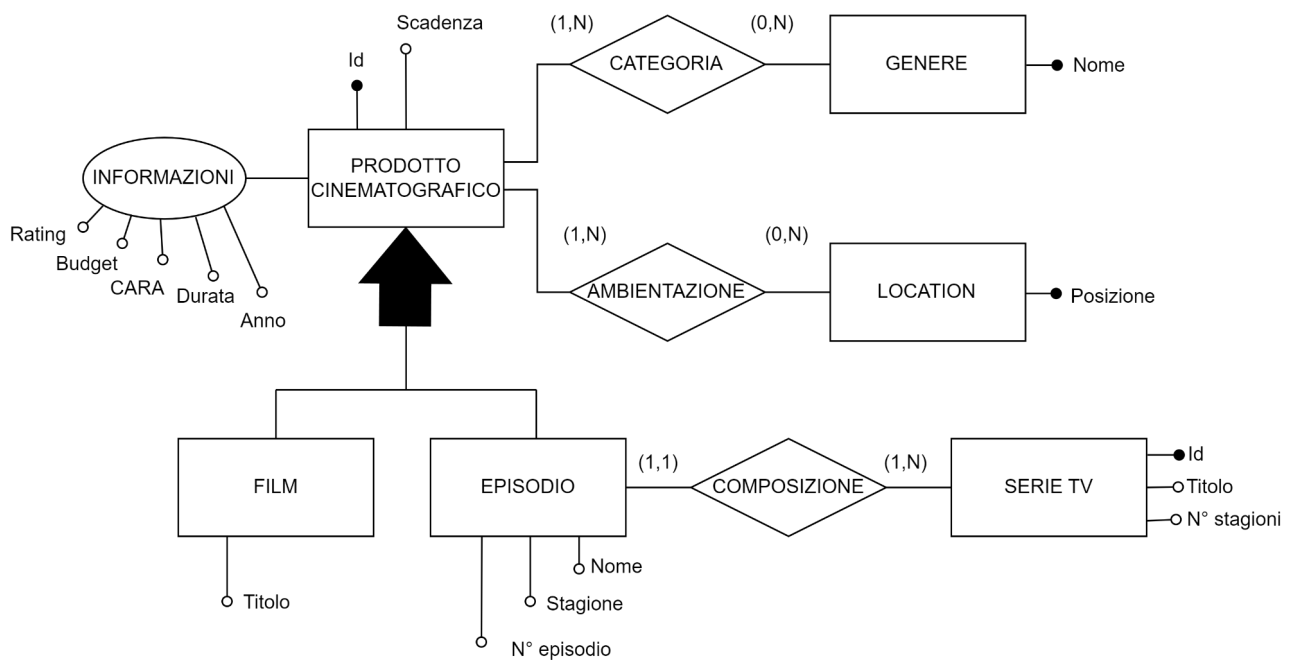


Abbiamo usato la disgiunzione totale in quanto un film non può essere una serie tv e viceversa. Inoltre tutti gli attributi del PRODOTTO CINEMATOGRAFICO vengono ereditati da entrambe le entità figlie.

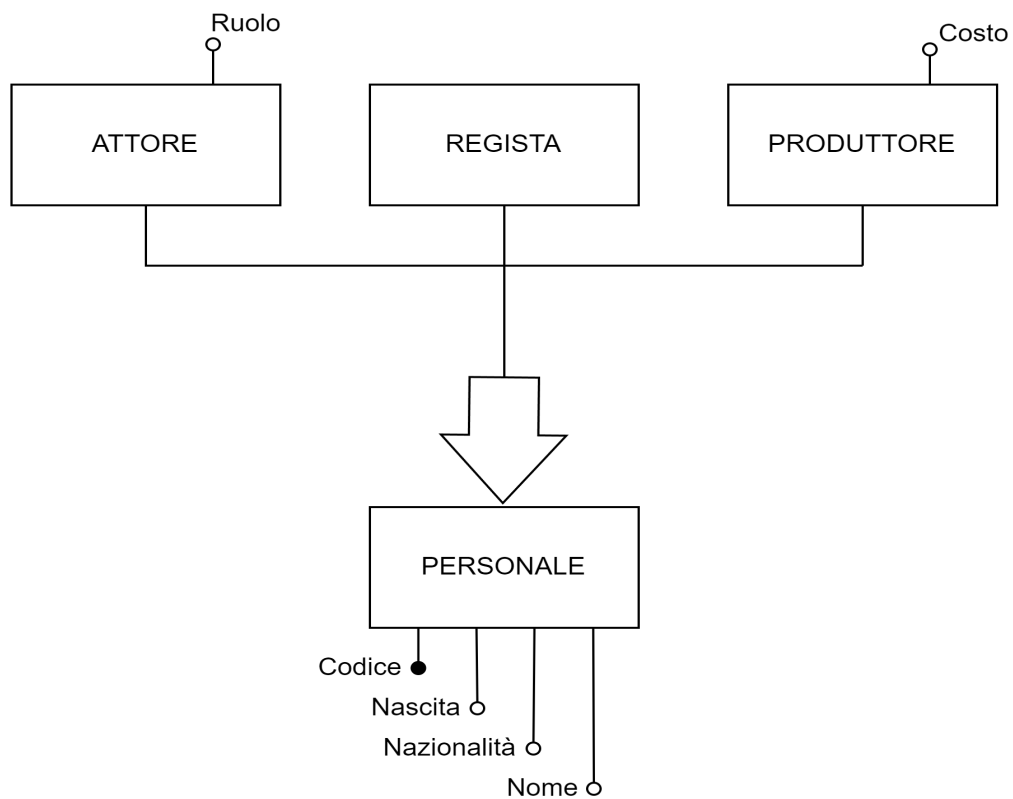
COMPOSIZIONE consente di associare un episodio alla sua serie tv di cui fa parte. Un episodio può appartenere a una sola serie tv, ma una serie tv è composta da almeno un episodio.

Abbiamo deciso di effettuare la reificazione degli attributi GENERE e LOCATION in quanto possono esistere autonomamente, inoltre un prodotto cinematografico può avere più generi e più location.

A seguito delle modifiche lo schema risulta essere:



### 2.5.3 Personale





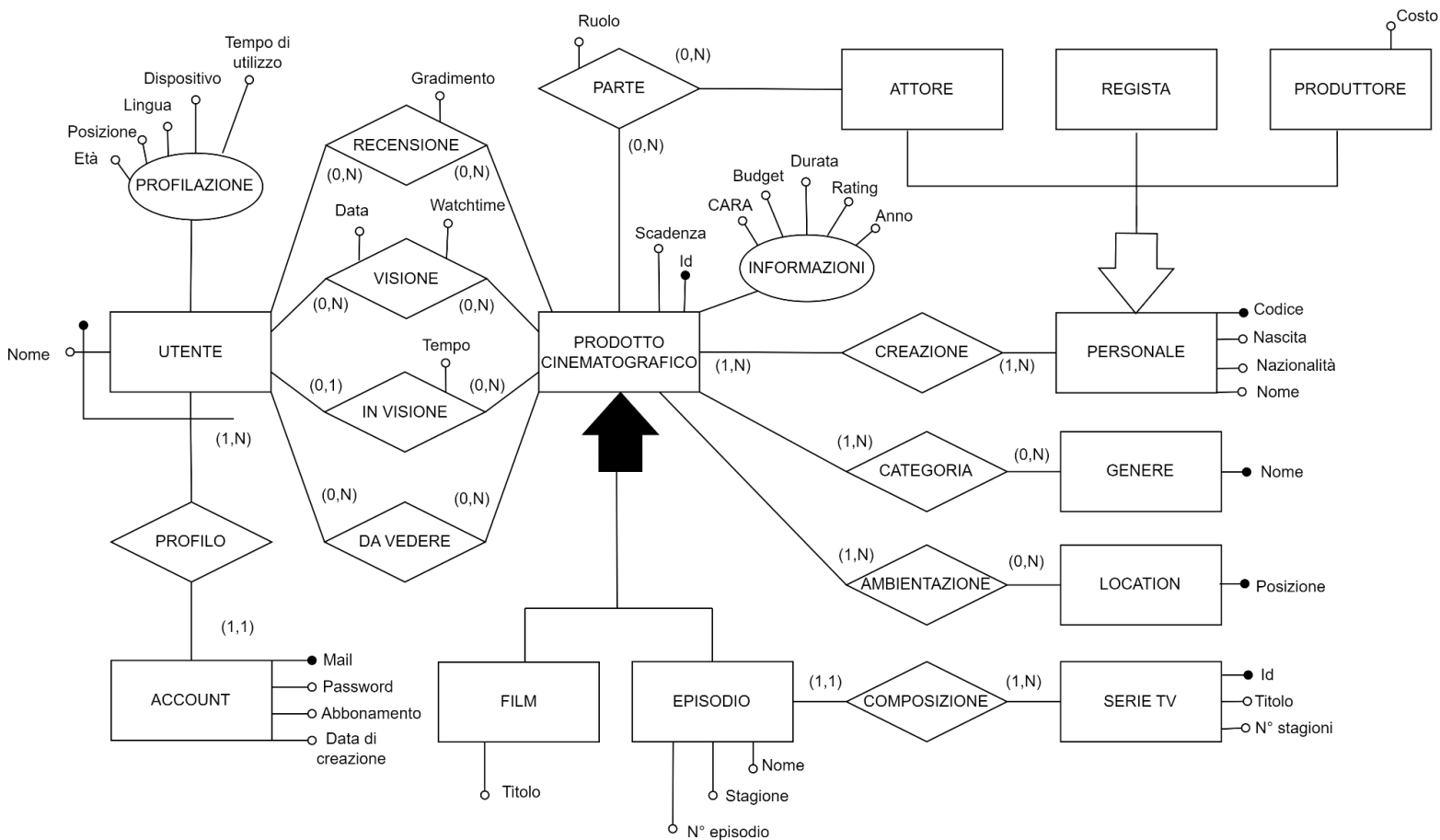
## 2.6 Unione componenti

Tra UTENTE e PRODOTTO CINEMATOGRAFICO abbiamo creato 4 relazioni: 3 delle quali per la gestione della visualizzazione dei film. VISIONATO, per i film già visti in passato, IN VISIONE per i film o serie attualmente in riproduzione e DA VEDERE, per i prodotti che si desiderano vedere in futuro.

La relazione RECENSIONE serve all'utente per esprimere opzionalmente una valutazione a seguito della visione del film.

La relazione PARTE associa l'ATTORE al ruolo svolto nel PRODOTTO CINEMATOGRAFICO, nel prodotto ci possono essere N parti e un attore può svolgere anche più parti nello stesso film.

Il processo di progettazione concettuale ci ha consentito di espandere le nostre 3 entità iniziali in un totale di 12, collegate tra loro da 10 relazioni, per rappresentare in maniera più fedele i requisiti progettuali.



## 2.1. Dizionario dei dati

### Entità:

Nome	Descrizione	Attributi	Identificatore
prodotto cinematografico	prodotto a disposizione degli utenti	anno, durata budget, rating, CARA, scadenza	id
film	tipo di prodotto autoconclusivo	titolo	-
episodio	tipo di prodotto seriale	nome, stagione, N°episodio	-
serie TV	insieme di episodi	titolo, n° stagioni	titolo
Genere	genere del prodotto		nome
Location	luoghi in cui è stato girato un prodotto		posizione
account	account con un abbonamento attivo	abbonamento, password, creazione	mail
utente	persona digitale che fa parte di un account	nome, profilazione	account, nome
personale	persone che hanno lavorato ai prodotti	nome, data di nascita, nazionalità	codice
regista	responsabile della direzione di un prodotto		-
attore	persona che recita in un prodotto		-
produttore	finanziatore di un prodotto	costo	

**Associazioni:**

Nome	Descrizione	Entità	Attributi
profilo	associa un utente all'account di riferimento	account (1,1) utente (1,N)	-
recensione	associa un utente al prodotto recensito	utente (0, N) prodotto (0, N)	gradimento
visione	associa un utente al prodotto visionato	utente (0, N) prodotto (0, N)	watchtime
in visione	associa un utente al prodotto che sta guardando	utente (0, 1) prodotto (0, N)	tempo
da vedere	associa un utente al prodotto che vorrà guardare in seguito	utente (0, N) prodotto (0, N)	-
composizione	associa un episodio alla serie TV di cui fa parte	episodio (1, 1) serie TV (1, N)	-
creazione	associa un lavoratore al prodotto a cui ha lavorato	prodotto (1, N) personale (1, N)	-
parte	associa un attore al prodotto nel quale ricopre una parte	prodotto (0, N) attore (0, N)	ruolo
Categoria	associa un prodotto ad un suo genere	prodotto (1, N) genere (0,N)	-
Ambientazioni	associa un prodotto ad una sua location	prodotto (1, N) location (0, N)	-

**2.2. Regole aziendali**

<b>Regole di vincolo</b>
(RV1) Ad ogni Abbonamento deve essere associato un tipo: <ul style="list-style-type: none"> <li>- mensile</li> <li>- semestrale</li> <li>- annuale</li> <li>- annuale PRO</li> </ul> (RV2) La Durata di un Prodotto cinematografico deve essere > 0 (RV3) Il Numero di Stagione di un Episodio deve essere > 0 (RV4) Il Tempo di un prodotto In Visione deve essere > 0 (RV5) Il Rating di un Prodotto cinematografico deve essere >= 0 e <= 10 (RV6) Il Budget di un Prodotto cinematografico deve essere > 0 (RV7) Il Gradimento di una Recensione deve essere >= 0 e <= 10
<b>Regole di derivazione</b>
(RD1) Il n° di stagioni della Serie TV è = max stagione di quella serie in Episodio

### 3. Progettazione logica

#### 3.1. Tabella dei volumi

Stima: 5000 prodotti e 1000 account:

Concetto	Tipo	Volume
Serie TV	E	150
Episodio	E	3500
Film	E	1500
Prodotto Cinematografico	E	5000
Location	E	10000
Genere	E	35
Produttore	E	600
Regista	E	600
Attore	E	50000
Personale	E	80000
Account	E	1000
Utente	E	2000
Composizione	R	3500
Creazione	R	100000
Parte	R	70000
Profilo	R	2000
Recensione	R	200
Visionato	R	200000
In Visione	R	500
Da Vedere	R	15000
Categoria	R	10000
Ambientazione	R	50000

### 3.2. Tabella delle operazioni

Operazione	Tipo	Frequenza
1	Interattiva	4 / sett
2	Batch	2 / sett
3	Interattiva	2 / sett
4	Batch	1 / giorno
5	Interattiva	5 / giorno
6	Interattiva	1 / giorno
7	Interattiva	1 / sett
8	Interattiva	6 / giorno
9	Interattiva	10 / giorno
10	Interattiva	10 / giorno
11	Interattiva	50 / giorno
12	Interattiva	50 / giorno
13	Interattiva	150 / giorno
14	Interattiva	5 / giorno
15	Interattiva	10 / giorno

### 3.3. Ristrutturazione dello schema concettuale

#### 3.3.1. Rimozione delle ridondanze

Abbiamo valutato di inserire un'altra possibile ridondanza per facilitare l'operazione del calcolo dei top 10 prodotti, aggiungendo un attributo "Visualizzazioni" nell'entità PRODOTTO CINEMATOGRAFICO.

Effettuare il calcolo della top 10 avviene per ogni prodotto, cercando tra i visionati quelli che compaiono più volte, di conseguenza il numero di accessi, avendo 200000 volumi nella relazione VISIONATO e 5000 PRODOTTI CINEMATOGRAFICI, in media si effettuano 40 accessi per prodotto.

#### Inserendo attributo Visualizzazioni, con ridondanza

Operazione 9: Ricerca della top ten dei prodotti visionati (10 volte al giorno)

Concetto	Costrutto	Accessi	Tipo
Prodotto	E	1	L

Operazione 12: Visione di un prodotto (150 volte al giorno)

Concetto	Costrutto	Accessi	Tipo
Prodotto	E	1	S
Visione	R	1	S

### Visualizzazioni, senza ridondanza

Operazione 9:

Concetto	Costrutto	Accessi	Tipo
Prodotto	E	1	L
Visione	R	40	L

Operazione 12:

Concetto	Costrutto	Accessi	Tipo
Prodotto	E	1	S

Costo

operazione 9

con ridondanza = 1 accesso \* 10 volte al giorno = 10

senza ridondanza = 41 accessi \* 10 volte al giorno = 410

operazione 12

con ridondanza = 2 \* 150 volte al giorno \* 2 perché è scrittura = 600

senza ridondanza = 1 \* 150 volte al giorno \* 2 perché è scrittura = 300

Costo totale con ridondanza = 10 + 600 = 610

Costo totale senza ridondanza = 410 + 300 = 710

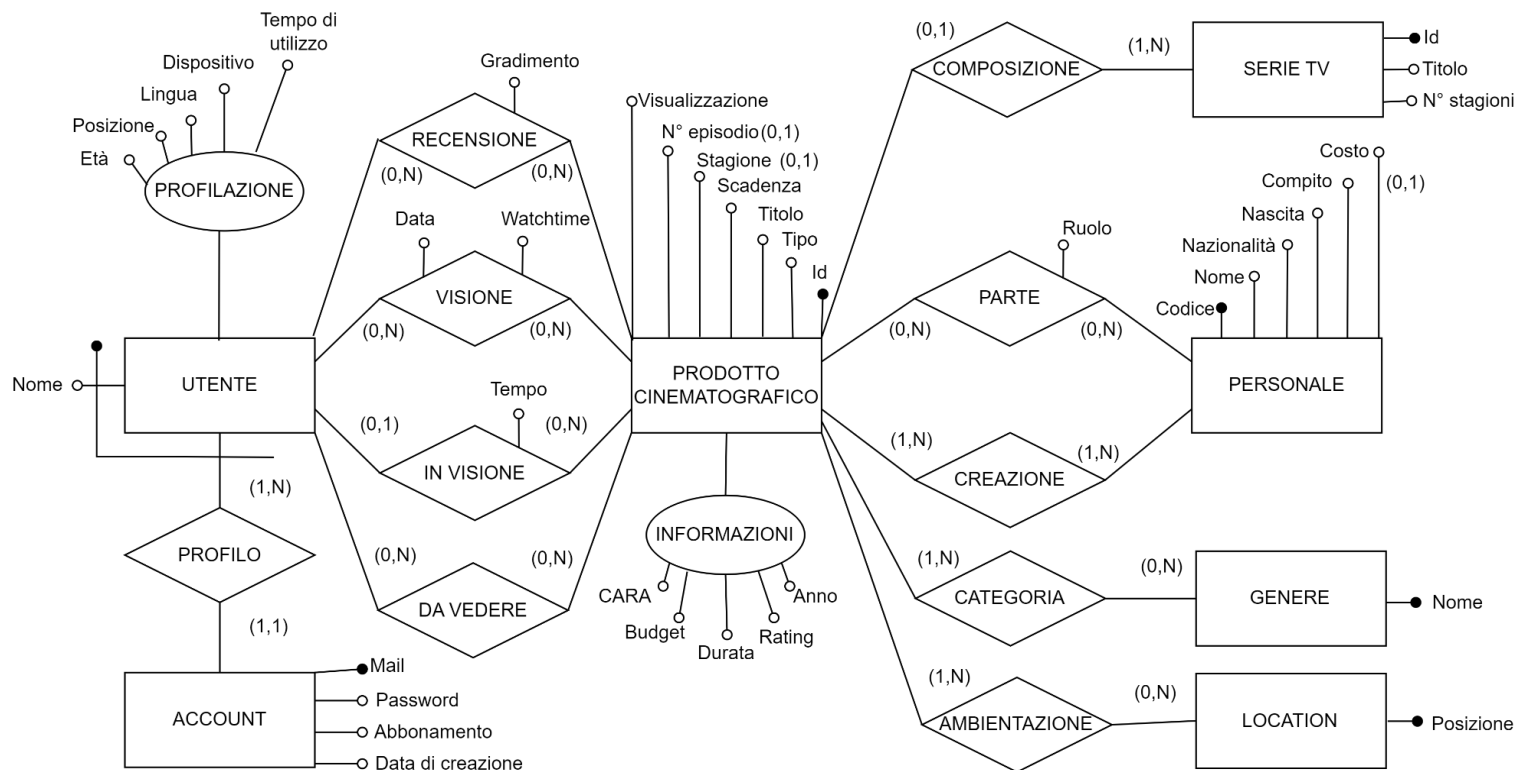
È più efficiente aggiungere la ridondanza.

Abbiamo rilevato una possibile ridondanza nell'attributo Stagione e Serie\_tv, in quanto tutti gli episodi di una stagione della stessa serie tv avranno gli stessi campi. Una soluzione potrebbe dunque essere quella di suddividere PRODOTTO CINEMATOGRAFICO in due entità separate, ma così facendo si creerebbe una ridondanza maggiore di quella iniziale. In quanto tutte le altre relazioni che coinvolgono i prodotti verrebbero duplicate, appesantendo il calcolo di tutte le operazioni di ricerca. Quindi abbiamo scelto di lasciare invariato lo schema.

### 3.3.2. Rimozione delle gerarchie

Nel nostro caso abbiamo 2 gerarchie da rimuovere:

1. **PRODOTTO CINEMATOGRAFICO** che ha 2 entità figlie: **FILM** ed **EPISODIO**  
la eliminiamo mettendo le entità figlie nell'entità padre, in quanto è probabile che si faccia l'accesso contemporaneamente
2. **PERSONALE** che ha 3 entità figlie: **ATTORE**, **REGISTA**, **PRODUTTORE**  
la eliminiamo allo stesso modo della gerarchia precedente



La cardinalità sinistra di **COMPOSIZIONE** è cambiata a (0,1) perchè nel caso si tratti di un film, non sarà associata nessuna **SERIE TV**, inoltre si noti l'aggiunta degli attributi:

“Tipo”, per discriminare tra serie tv e film,

“Titolo”, che raggruppa gli attributi “Titolo” di **FILM** e “Nome” di **Serie TV**

“Stagione”, è accompagnato dalla molteplicità (0,1), essendo specifico per l'entità figlia **SERIE TV**.

In maniera analoga, sono stati aggiunti all'entità **PERSONALE** i seguenti attributi:

“Compito”, per diversificare i ruoli di Attore, Regista, Produttore e Altro,

“Costo”, è accompagnato dalla molteplicità (0,1), essendo specifico per l'entità figlia **PRODUTTORE**.

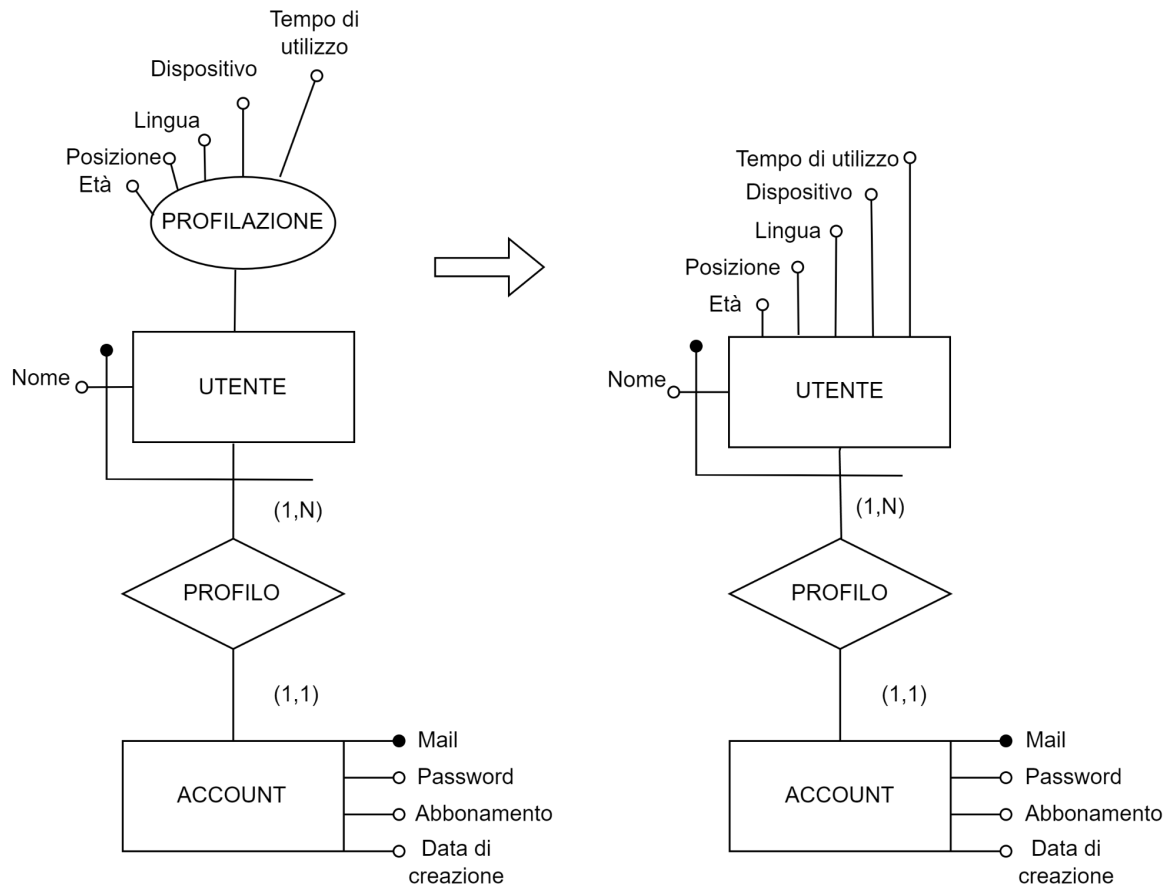
Durante questa fase abbiamo valutato di effettuare una partizione verticale degli attributi **PRODOTTO CINEMATOGRAFICO** e **UTENTE**, ma visto che

l'operazione creerebbe ridondanza, in ultima istanza abbiamo preferito lasciare lo schema invariato.

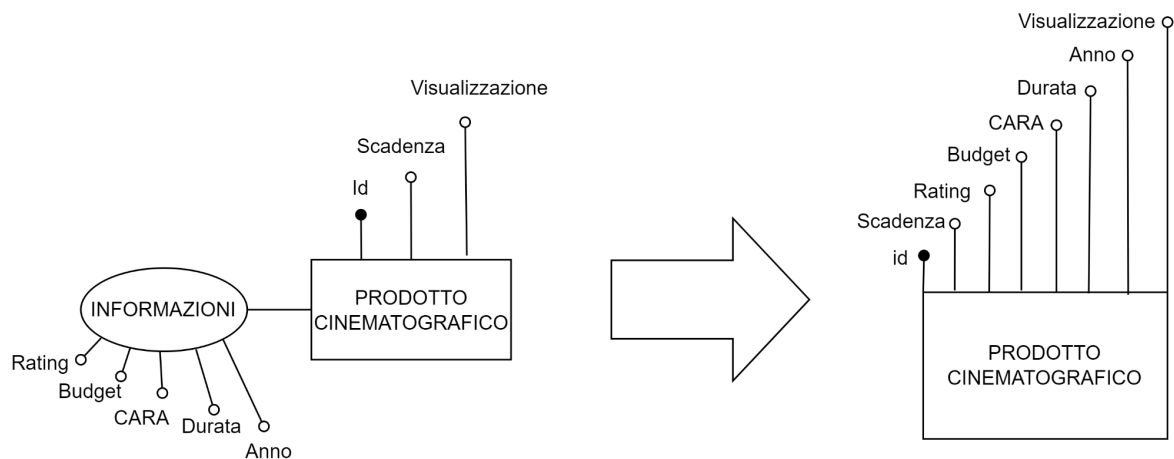
### 3.3.3. Accorpamento

L'accorpamento si è verificato come conseguenza dell'eliminazione delle gerarchie nell'entità PERSONALE e PRODOTTO CINEMATOGRAFICO

### 3.3.4. Rimozione degli attributi multivalore



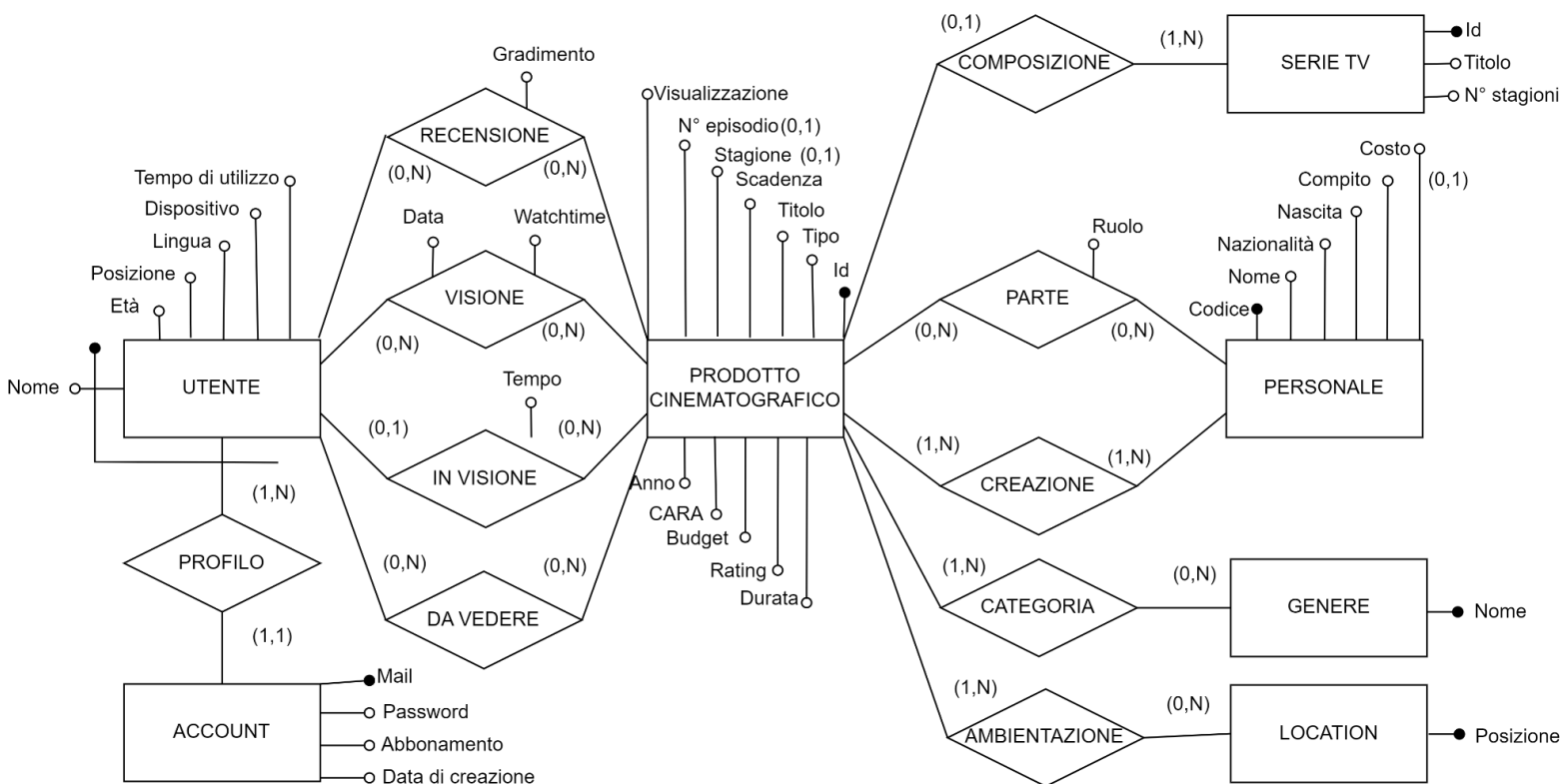




### 3.3.5. Identificazione delle chiavi primarie

Le chiavi primarie sono state associate a ciascuna identità nello schema concettuale, pertanto non occorre effettuare modifiche sullo schema ER per questo punto.

**Lo schema ER alla fine della ristrutturazione risulta essere:**



### 3.4. Normalizzazione

#### Associazioni

Sono già normalizzate in BCNF perchè sono tutte binarie.

#### Entità

Non essendoci dipendenze non banali tra gli attributi, le entità sono tutte in BCNF

### 3.5. Traduzione nel modello relazionale

Entità / Relazione	Traduzione
Serie TV	SerieTV( <u>Id</u> , Titolo, NStagioni)
Prodotto cinematografico	ProdCinema( <u>Id</u> , Rating, Durata, Budget, Anno, Titolo, CARA, Scadenza, Tipo, Stagione, SerieTV, NumEpisodio, Visual)
Personale	Personale( <u>Codice</u> , Nome, DataNasc, Naz)
Account	Account( <u>Mail</u> , Pwd, Abbonamento, DataCreaz)
Utente	Utente( <u>Nome</u> , <u>Account</u> , Età, Posizione, Ling, Disp, TempoUtilizzo)
Ambientazione	Ambientazione(ProdCinema, Location)
Categoria	Categoria(ProdCinema, Genere)
Creazione	Creazione(ProdCinema, Personale)
Parte	Parte(ProdCinema, Attore, Ruolo)
Recensione	Recensione( <u>Utente</u> , <u>Account</u> , <u>ProdCinema</u> , Grad)
Visione	Visionato( <u>Utente</u> , <u>Account</u> , <u>ProdCinema</u> , WatchTime, Data)
InVisione	In visione( <u>Utente</u> , <u>Account</u> , <u>ProdCinema</u> , Tempo)
Da vedere	Da vedere( <u>Utente</u> , <u>Account</u> , <u>ProdCinema</u> )

Traduzione	Vincoli di riferimento
SerieTV( <u>Id</u> , Titolo, NStagioni)	-
ProdCinema( <u>Id</u> , Rating, Durata, Budget, Anno, Titolo, CARA, scadenza, Tipo, Stagione, SerieTV, NumEpisodio, Visual)	SerieTV → SerieTV.Id

Personale( <u>Codice</u> , Nome, DataNasc, Naz, Compito)	-
Account( <u>Mail</u> , Pwd, Abbonamento, DataCreaz)	-
Utente( <u>Nome</u> , <u>Account</u> , Età, Posizione, Ling, Disp, TempoUtilizzo)	Account → Account.mail
Ambientazione(ProdCin <u>ma</u> , Location)	ProdCinma → ProdCinma.Id
Categoria(ProdCin <u>ma</u> , Genere)	ProdCinma → ProdCinma.Id
Creazione( <u>ProdCinma</u> , <u>Personale</u> )	ProdCinma → ProdCinma.Id Personale → Personale.Codice
Parte( <u>ProdCinma</u> , <u>Attore</u> , <u>Ruolo</u> )	ProdCinma → ProdCinma.Id Attore → Personale.Codice
Recensione( <u>Utente</u> , <u>Account</u> , <u>ProdCinema</u> , Grad)	Utente → Utente.Nome Account → Account.Mail ProdCinma → ProdCinma.Id
Visione( <u>Utente</u> , <u>Account</u> , <u>ProdCinema</u> , WatchTime, Data)	Utente → Utente.Nome Account → Account.Mail ProdCinma → ProdCinma.Id
InVisione( <u>Utente</u> , <u>Account</u> , <u>ProdCinema</u> , Tempo)	Utente → Utente.Nome Account → Account.Mail ProdCinma → ProdCinma.Id
Da vedere( <u>Utente</u> , <u>Account</u> , <u>ProdCinema</u> )	Utente → Utente.Nome Account → Account.Mail ProdCinema → ProdCinema.Id

## 4. Codifica SQL

### 4.1. Definizione dello schema

```
CREATE DATABASE riverflix
```

```
CREATE TABLE SerieTV(  
    Id MEDIUMINT UNSIGNED not NULL AUTO_INCREMENT,  
    Titolo VARCHAR(30) not NULL,  
    NStagioni SMALLINT not NULL,  
    PRIMARY KEY (Id)  
)
```

```
CREATE TABLE ProdCinema(  
    Id MEDIUMINT UNSIGNED not NULL AUTO_INCREMENT,  
    Rating SMALLINT DEFAULT 0,  
    Durata MEDIUMINT not NULL,  
    Budget INT not NULL,  
    Anno YEAR(4),  
    Titolo VARCHAR(30) not NULL,  
    Cara ENUM('G','PG','PG-13','R','NC-17'),  
    Scadenza DATE not NULL,  
    Tipo ENUM('serie_tv','film'),  
    Stagione SMALLINT,  
    NumEpisodio SMALLINT,  
    Serietv MEDIUMINT UNSIGNED,  
    Visual INT DEFAULT 0,  
  
    PRIMARY KEY (Id)  
  
    FOREIGN KEY (Serietv)  
        REFERENCES SerieTV(Id)  
  
    CHECK(Rating >= 0 AND Rating <= 10)  
)
```

```
CREATE TABLE Personale(  
    Codice CHAR(16) not NULL,  
    Nome VARCHAR(20) not NULL,  
    DataNasc DATE,  
    Nazionalità VARCHAR(30),  
    Compito VARCHAR(20),  
  
    PRIMARY KEY(Codice)  
)
```

```
CREATE TABLE Account(  
    Mail VARCHAR(40) not NULL,  
    Password VARCHAR(100) not NULL,  
    Abbonamento ENUM('mensile', 'semestrale', 'annuale', 'annuale  
PRO') not NULL,  
    DataCreaz DATE not NULL
```

```

PRIMARY KEY (Mail)
CHECK (Mail REGEXP "^[\\w-\\.]+@([\\w-]+\\.){2,4}$")
)

CREATE TABLE Utente(
    Nome VARCHAR(20) not NULL,
    Account VARCHAR(40) not NULL,
    Età SMALLINT,
    Posizione VARCHAR(100)
    Ling VARCHAR(30),
    Dispositivo VARCHAR(20)
    TempoUtilizzo MEDIUM INT UNSIGNED

    PRIMARY KEY (Nome, Account)

    FOREIGN KEY (Account)
        REFERENCES Account (Mail)
)

CREATE TABLE Ambientazione(
    ProdCinema MEDIUMINT UNSIGNED not NULL,
    Location VARCHAR(30),

    FOREIGN KEY (ProdCin)
        REFERENCES ProdCinema (Id)
)

CREATE TABLE Categoria(
    ProdCinema MEDIUMINT UNSIGNED not NULL,
    Genere VARCHAR(30),

    FOREIGN KEY (ProdCin)
        REFERENCES ProdCinema (Id)
)

CREATE TABLE Creazione(
    ProdCinema MEDIUMINT UNSIGNED not NULL,
    Oersonale CHAR(16) not NULL,

    FOREIGN KEY (ProdCin)
        REFERENCES ProdCinema (Id)
    FOREIGN KEY (Personale)
        REFERENCES Personale (Codice)
)

CREATE TABLE Parte(
    ProdCinema MEDIUMINT UNSIGNED not NULL,
    Attore CHAR(16) not NULL,
    Ruolo VARCHAR(20) not NULL,

    PRIMARY KEY (ProdCinema , Attore, Ruolo)

    FOREIGN KEY (ProdCinema)
        REFERENCES ProdCinema (Id)
    FOREIGN KEY (Attore)
        REFERENCES Personale (Codice)
)

```

)

```
CREATE TABLE Recensione(  
    Utente VARCHAR(20) not NULL,  
    Account VARCHAR(40) not NULL,  
    ProdCinema MEDIUMINT UNSIGNED not NULL,  
    Gradimento SMALLINT not NULL,  
  
    PRIMARY KEY (Utente, Account, ProdCinema )  
  
    FOREIGN KEY (Utente)  
        REFERENCES Utente(Nome)  
    FOREIGN KEY (Account)  
        REFERENCES Account(Mail)  
    FOREIGN KEY (ProdCinema)  
        REFERENCES ProdCinema(Id)  
  
    CHECK(Gradimento <= 10 AND Gradimento >= 0)  
)
```

```
CREATE TABLE Visione(  
    Utente VARCHAR(20) not NULL,  
    Account VARCHAR(40) not NULL,  
    ProdCinema MEDIUMINT UNSIGNED not NULL,  
    Watchtime INT UNSIGNED not NULL,  
    Data DATE not NULL  
  
    PRIMARY KEY (Utente, Account, ProdCinema )  
  
    FOREIGN KEY (Utente)  
        REFERENCES Utente(Nome)  
    FOREIGN KEY (Account)  
        REFERENCES Account(Mail)  
    FOREIGN KEY (ProdCinema )  
        REFERENCES ProdCinema(Id)  
)
```

```
CREATE TABLE InVisione(  
    Utente VARCHAR(20) not NULL,  
    Account VARCHAR(40) not NULL,  
    ProdCinema MEDIUMINT UNSIGNED not NULL,  
    Tempo INT UNSIGNED not NULL,  
  
    PRIMARY KEY (Utente, Account, ProdCinema )  
  
    FOREIGN KEY (Utente)  
        REFERENCES Utente(Nome)  
    FOREIGN KEY (account)  
        REFERENCES Account(Mail)  
    FOREIGN KEY (ProdCinema )  
        REFERENCES ProdCinema(Id)  
)
```

```
CREATE TABLE DaVedere(  
    Utente VARCHAR(20) not NULL,  
    Account VARCHAR(40) not NULL,
```

```

        ProdCinema MEDIUMINT UNSIGNED not NULL,

        PRIMARY KEY (Utente, Account, ProdCinema )

        FOREIGN KEY (Utente)
            REFERENCES Utente(Nome)
        FOREIGN KEY (Account)
            REFERENCES Account(Mail)
        FOREIGN KEY (ProdCinema )
            REFERENCES ProdCinema(Id)
    )

```

## 4.2. Codifica delle operazioni

### 4.2.1. .Aggiunta di un prodotto cinematografico (in media 4 volta a settimana)

```

INSERT INTO ProdCinema (Id, Rating, Durata, Budget, Anno,
    Titolo, CARA, Scadenza, Tipo, Stagione, SerieTV,
    NumEpisodio)
VALUES (...)

```

per tutto il personale:

```

INSERT INTO Personale(Codice, Nome, DataNasc, Nazionalità,
    Compito)
VALUES (...)

```

```

INSERT INTO Creazione(ProdCinema, Personale)
VALUES (...)

```

per ogni attore:

```

INSERT INTO Parte(ProdCinema, Attore, Ruolo)
VALUES (...)

```

per ogni categoria:

```

INSERT INTO Categoria(ProdCinema, Genere)
VALUES (...)

```

per ogni location:

```

INSERT INTO Ambientazione(ProdCinema, Location)
VALUES (...)

```

### 4.2.2. Rimozione di un prodotto cinematografico (in media 2 volte a settimana)

```

DELETE FROM ProdCinema
WHERE Scadenza < CAST(GETDATE() AS Date)

```

### 4.2.3. Aggiornamento di un prodotto cinematografico (in media 2 volta a settimana)

```

UPDATE ProdCinema
SET ...
WHERE Id = ...

```

**4.2.4. Aggiornamento del rating di un prodotto in base alle recensioni (1 volta al giorno)**

```
UPDATE ProdCinema
SET Rating = t.meanR
FROM (
    SELECT ProdCinema, AVG(ProdCinema) as meanR
    FROM Recensione
    GROUP BY ProdCinema
) t
WHERE Id = t.ProdCinema
```

**4.2.5. Aggiunta di un account (in media 5 volte al giorno)**

```
INSERT INTO Account (Mail, Psw, Abbonamento, DataCreaz)
VALUES (...)
```

**4.2.6. Cambio abbonamento (in media 1 volta al giorno)**

```
UPDATE Account
SET Abbonamento = '...'
WHERE Mail = '...'
```

**4.2.7. Rimozione di un account (in media 1 volta a settimana)**

```
DELETE FROM Account
WHERE Mail = '...'
```

**4.2.8. Aggiunta di un utente (6 volte al giorno)**

```
INSERT INTO Utente (Nome, Account, Età, Posizione, Ling,
    Disp, TempoUtilizzo)
VALUES (...)
```

**4.2.9. Ricerca top 10 prodotti più visti (in media 10 volte al giorno)**

```
SELECT Id, Titolo, Tipo
FROM ProdCinema
ORDER BY COUNT(Visual) DESC
LIMIT 10
```

**4.2.10. Ricerca della top ten dei prodotti visionati in questo mese (in media 10 volte al giorno)**

```
SELECT P.Id, P.Titolo, P.Tipo
FROM ProdCinema as P JOIN Visione as V on
    P.Id = V.ProdCinema
WHERE V.Data > DATE_ADD((CAST(GETDATE() AS Date) INTERVAL -1
    MONTH)
GROUP BY V.ProdCinema
ORDER BY COUNT(V.ProdCinema) DESC
LIMIT 10
```



- 4.2.11. Ricerca prodotti consigliati in base alle visioni precedenti (in media 100 volte al giorno)  
È una semplificazione, ma assumiamo che gli utenti vogliano prodotti dello stesso genere di quelli già visti

```
SELECT P.Id, P.Titolo, P.Tipo
FROM Visione as V JOIN ProdCinema as P ON V.ProdCinema = P.Id
JOIN Categoria as C ON P.Id = C.ProdCin JOIN (
    SELECT C.genere as FavGen
    FROM Visione as V JOIN Categoria as C ON
        V.ProdCinema = C.ProdCin
    WHERE V.Utente = 'Federica' AND V.Account =
        'famiglia@fam.com' GROUP BY C.Genere
    ORDER BY COUNT(*) DESC
    LIMIT 1
) t ON C.Genere = t.FavGen
LIMIT 5
```

- 4.2.12. Ricerca di un prodotto (in media 50 volte al giorno)

```
SELECT P.Id, P.Titolo, P.Tipo
FROM ProdCinema
WHERE ...
```

- 4.2.13. Visione di un prodotto da parte di un utente (in media 150 volte al giorno)

```
INSERT INTO Visionato (Utente, Account, ProdCinema,
Watchtime)
VALUES (...)
```

- 4.2.14. Valutazione di un prodotto da parte di un utente (in media 5 volte al giorno)

```
INSERT INTO Recensione (Utente, Account, ProdCinema,
Gradimento)
VALUE (...)
```

- 4.2.15. Ricerca delle informazioni di un prodotto (in media 10 volte al giorno)

```
SELECT Rating, Durata, Budget, Anno, CARA, Stagione, SerieTV
FROM ProdCinema
WHERE ...
```

## 5. Testing

Codice per testare il DB:

<https://github.com/mattyk0207/progDB>

Risultato delle operazioni:

<https://github.com/mattyk0207/progDB/tree/main/DBdump>