

Progetto di Basi di Dati - Sito Cinema e TV

Samuele Pecetto

July 22, 2020

Contents

1	Progettazione concettuale	3
1.1	Requisiti iniziali	3
1.2	Glossario dei termini	3
1.3	Requisiti rivisti e strutturati in gruppi di frasi omogenee	4
1.4	Schema E-R + business rules	4
2	Progettazione logica	7
2.1	Tavola dei volumi	7
2.2	Tavola delle operazioni	8
2.3	Ristrutturazione schema E-R	9
2.3.1	Analisi delle ridondanze	9
2.3.2	Eliminazione delle generalizzazioni	10
2.3.3	Partizionamento/Accorpamento di entità e associazioni	11
2.3.4	Scelta degli identificatori principali	11
2.4	Schema E-R ristrutturato + business rules	12
2.5	Schema relazionale	13
3	Implementazione	15
3.1	DDL: Creazione del Database	15
3.2	DML: Popolamento tabelle	19
3.3	Constraints check: Operazioni di cancellazione e modifica	31

1 Progettazione concettuale

1.1 Requisiti iniziali

Si veda il documento *esame_progettazione_requisiti.pdf* contenente l'analisi funzionale del progetto.

1.2 Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Redattore	Utente responsabile del caricamento e aggiornamento dei contenuti disponibili sulla piattaforma	Utente redazione	Contenuto, Metadati
Metadati	Informazioni inserite dai redattori, relative ai contenuti come Titolo, Paese, Genere, Data di uscita ecc.	dati dei Film, Serie, ecc	Contenuto, Redattore
Visitatore	Utente registrato capace di votare e/o aggiungere ai propri preferiti, contenuti pubblicati dai redattori	Utente	Preferiti, Voto
Voto	Votazione espressa con un numero naturale compreso tra 1 e 5 da un utente Visitatore per un contenuto	Votazione	Contenuto, Utente, Visitatore
Preferiti	Insieme di contenuti che il Visitatore può registrare nella sua area personale		Visitatore
Contenuti	Insieme di risorse cinematografiche	Film, Serie, Episodi, Programmi	
Piattaforma	Mezzo di trasmissione dei contenuti - Streaming, canali DTV		Serie, Programmi
Sito	Sito internet dell'applicazione tipo Coming Soon	Sito Cinema e TV	
Personaggio	Individuo appartenente al mondo dello spettacolo. Vengono considerati Personaggi i Registi, Sceneggiatori, Attori, Fotografi, Musicisti e più in generale chiunque partecipi attivamente alla realizzazione di contenuti cinematografici		Contenuto
Fotografo	Abbreviazione del più preciso Direttore della fotografia	Direttore della fotografia	Contenuto, Personaggio
Interpretazione	Nome del soggetto di fantasia interpretato dal personaggio nel contenuto specificato		Personaggio, Attore
Indirizzo	Aggregazione di Paese, Provincia, Città, Indirizzo e Civico di un luogo	Luogo Nascita	Cinema
Contatti	Metodi di contatto come numeri di telefono e/o indirizzi email	Contatti telefonici	Cinema

1.3 Requisiti rivisti e strutturati in gruppi di frasi omogenee

Si vuole realizzare una base di dati per la gestione di una sito che fornisce informazioni su contenuti cinematografici come film, serie TV e programmi TV, ispirato a siti internet come ComingSoon.

I redattori sono utenti responsabili dell'aggiornamento dei contenuti e metadati disponibili sul sito come Film, Serie TV e relativi Episodi, Programmi TV.

I redattori sono responsabili dell'inserimento delle programmazioni delle proiezioni dei Film nelle sale dei cinema per ogni contenuto disponibile sul sito.

I redattori sono responsabili dell'inserimento delle programmazioni televisive per i contenuti disponibili sul sito.

Sul sito possono registrarsi degli utenti visitatori che possono aggiungere nei loro preferiti e votare i contenuti presenti sul sito. Per i visitatori e per i redattori si devono memorizzare username, password e email.

Per i redattori si deve memorizzare anche la data di inizio collaborazione.

I contenuti hanno una serie di metadati da memorizzare che consistono di: Titolo, Descrizione, Data di uscita, Genere, Anno, Registi, Attori con relativa interpretazione, Paese, Durata, Distributori, Sceneggiatori, Fotografi, Musicisti e Produttori.

Ogni serie TV è composta da una o più stagioni.

Ogni stagione è composta da uno o più episodi, dei quali si devono memorizzare specifici metadati che consistono di: Titolo, Registi, Attori, Sceneggiatori, Durata.

Per ogni personaggio si deve memorizzare: Nome, Foto, Biografia, Contenuti in cui ha partecipato, Data e luogo di nascita.

I film vengono proiettati nei cinema. Per ogni cinema si deve memorizzare: Nome, Contatti telefonici, Indirizzo.

Di ogni proiezione si deve memorizzare: Data e ora, Sala del cinema, Prezzo del biglietto.

Programmi e Serie TV vanno in onda su canali TV o piattaforme di streaming video, dove vengono organizzate per genere.

1.4 Schema E-R + business rules

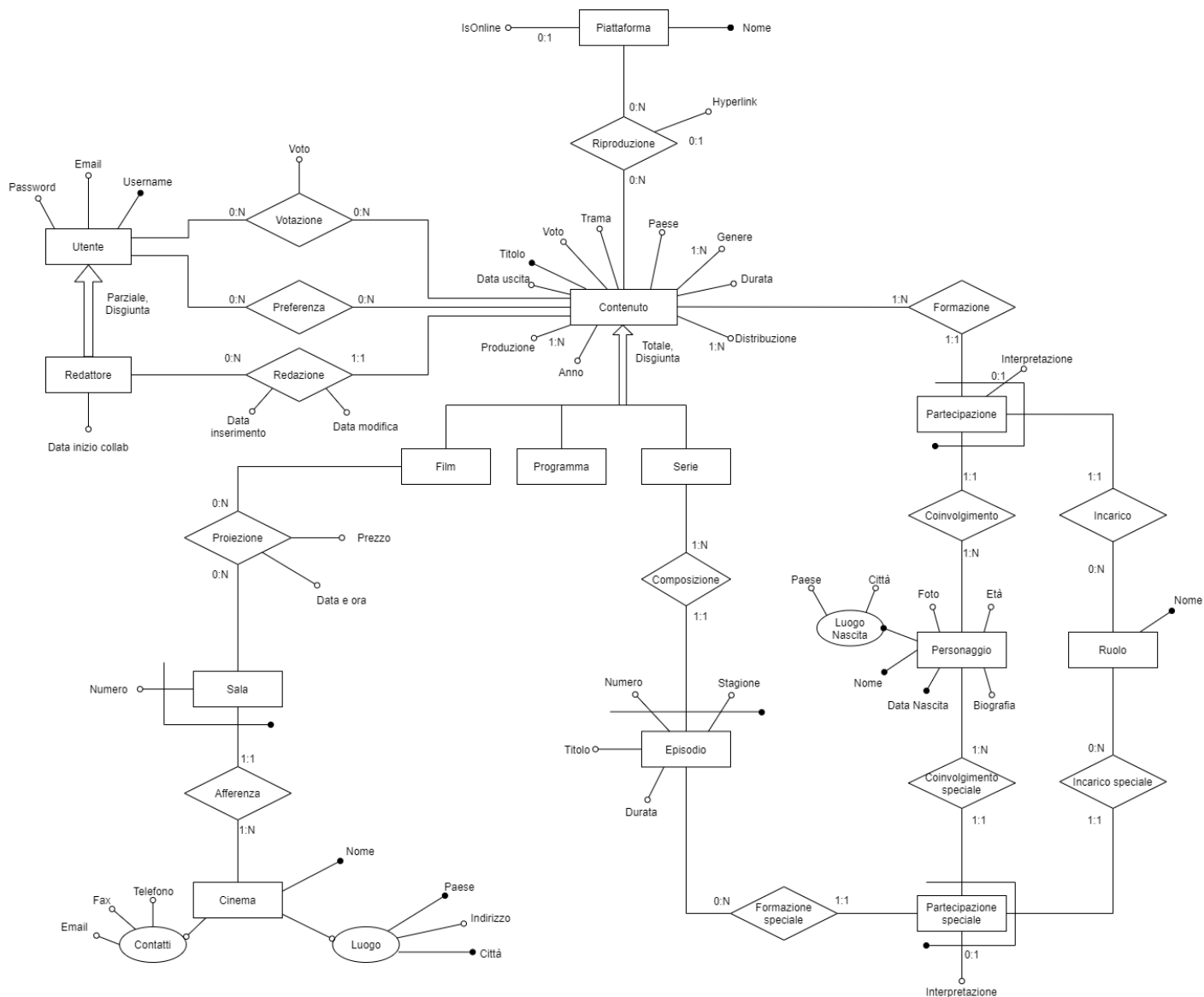


Figure 1: Schema ER iniziale

Osserviamo che l'identificatore primario delle entità deboli Partecipazione e Partecipazione speciale includono l'attributo opzionale Interpretazione, in quanto si vuole supportare la casistica in cui un Personaggio può far parte del cast di un Contenuto con svariati ruoli, oltre che come attore di diverse interpretazioni. Si pensi ad esempio a un film di Carlo Verdone dove interpreta diversi personaggi ed è anche regista dei suoi film.

Essendo però illegale la definizione di una chiave primaria composita con attributi opzionali, nella fase di ristrutturazione verrà introdotto un identificatore numerico sequenziale per ovviare al problema, permettendo la definizione di tuple che altrimenti creerebbero violazione di chiave, negli scenari appena descritti.

Regole di vincolo

- L'attributo Voto dell'associazione Votazione tra Utente e Contenuto è un numero naturale compreso tra 1 e 5.
- L'attributo Interpretazione dell'entità Partecipazione/Partecipazione speciale deve essere valorizzato soltanto nelle istanze con ruolo pari ad Attore.
- L'attributo Hyperlink dell'associazione tra Contenuto e Piattaforma può essere valorizzato solo nel caso in cui la Piattaforma sia online.
- L'attributo Data modifica dell'associazione Redazione tra Redattore e Contenuto deve essere strettamente maggiore della Data inserimento.

Regole di derivazione

- La Votazione di un contenuto è ottenuta facendo la media aritmetica dei voti per quel contenuto di tutti gli utenti.
- L'anno di un contenuto è ottenuto estraendo l'anno dalla data di uscita.
- Il cast di default di un Episodio è desumibile dal cast associato alla Serie di appartenenza.

2 Progettazione logica

2.1 Tavola dei volumi

Si suppone che ogni serie abbia mediamente 2 stagioni di cui ognuna composta da 8 episodi.

Si suppone che ogni episodio che ha specifici attori, registi e sceneggiatori, siano già partecipanti ad altri contenuti (non vanno a modificare la cardinalità di Personaggi).

Si suppone che Piattaforma contenga oltre ai siti di streaming (10), anche i canali TV (40).

Si suppone che mediamente per ogni contenuto ci siano 5 attori con ruoli principali di cui si vuole tenere traccia.

Si suppone che mediamente ogni attore partecipi a 5 contenuti.

Si suppone che mediamente ci sia 1 regista per ogni contenuto.

Si suppone che mediamente ogni regista partecipi a 10 contenuti.

Si suppone che mediamente ci siano 2 sceneggiatori per ogni contenuto.

Si suppone che mediamente ogni sceneggiatore partecipi a 10 contenuti.

Si suppone che mediamente ci sia 1 fotografo per ogni contenuto.

Si suppone che mediamente ogni fotografo partecipi a 20 contenuti.

Si suppone che mediamente ci sia 1 musicista per ogni contenuto.

Si suppone che mediamente ogni musicista partecipi a 50 contenuti.

Si suppone che mediamente ogni 10 episodi di una serie ci siano 1 regista, 1 attore, 1 sceneggiatore, 1 musicista e 1 fotografo diversi rispetto a quelli della serie stessa.

Si suppone che mediamente ogni cinema abbia 2 sale.

Si suppone che ogni utente voti mediamente 50 contenuti diversi.

Si suppone che ogni utente aggiunga ai preferiti mediamente 5 contenuti diversi.

Si suppone che ogni contenuto venga mediamente riprodotto su 5 piattaforme diverse.

Si suppone che ogni film venga riprodotto 2 volte in un cinema.

Concetto	Tipo	Volume
Utente	E	10000
Redattore	E	100
Contenuto	E	100000
Partecipazione	E	1000000
Partecipazione speciale	E	160000
Ruolo	E	5
Film	E	50000
Programma	E	30000
Serie	E	20000
Episodio	E	320000
Piattaforma	E	50
Personaggio	E	47000
Sala	E	600
Cinema	E	300
Votazione	R	500000
Preferenza	R	50000
Redazione	R	100000
Riproduzione	R	500000
Composizione	R	320000
Proiezione	R	30000000
Afferenza	R	600
Formazione	R	1000000
Formazione speciale	R	160000
Coinvolgimento	R	1000000
Coinvolgimento speciale	R	160000
Incarico	R	1000000
Incarico speciale	R	160000

2.2 Tavola delle operazioni

Basandoci sulla regola dell'80-20, consideriamo le operazioni di seguito elencate come le più costose, in termini di esecuzione e frequenti.

In particolare, il criterio di scelta delle seguenti operazioni è veicolato dalle entità e relazioni con il maggior numero di tuple coinvolte, che risultano essere quelle delle proiezioni e delle partecipazioni dei personaggi a contenuti.

Inoltre, si assume che lo scopo principale del sito sia quello di fornire capacità agli utenti di visualizzare metadati dei loro film/attori preferiti. Questo porta a scegliere prevalentemente operazioni interattive.

Operazione	Tipo	Frequenza
Registrazione nuovo Utente	I	20 / giorno
Redazione nuovo contenuto	I	20 / giorno
Ricerca contenuto	I	5000 / giorno
Ricerca personaggio	I	5000 / giorno
Visualizzazione scheda contenuto	I	5000 / giorno
Visualizzazione scheda personaggio	I	5000 / giorno
Votazione di un contenuto	I	500 / giorno
Ricerca di proiezioni disponibili per un film	I	500 / giorno
Aggiornamento media delle votazioni di un contenuto	B	500 / giorno

I: Interattiva **B:** Batch

2.3 Ristrutturazione schema E-R

2.3.1 Analisi delle ridondanze

Attributo Voto

Nell'attuale formalizzazione si possono riscontrare delle ridondanze nell'attributo Voto presente in Contenuto.

Infatti il valore medio si può estrarre tramite interrogazione nell'associazione Votazione tra Utente e Contenuto.

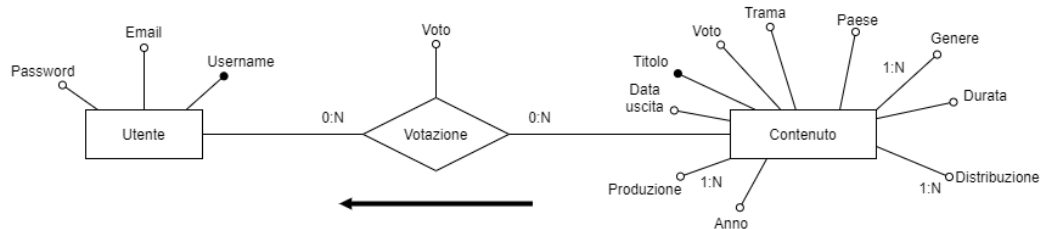


Figure 2: Schema di navigazione

Considerando pessimisticamente che un contenuto riceve mediamente 5 voti, sono necessari 5 accessi in lettura nella relazione Votazione per determinarne la media.

Concetto	Costrutto	Accessi	Tipo
Contenuto	Entità	1	R

Table 1: Tavola degli accessi dell'operazione Visualizzazione scheda contenuto **con ridondanza**

Concetto	Costrutto	Accessi	Tipo
Votazione	Relazione	4	R
Votazione	Relazione	1	W
Contenuto	Entità	1	W

Table 2: Tavola degli accessi dell'operazione votazione contenuto **con ridondanza**

Concetto	Costrutto	Accessi	Tipo
Contenuto	Entità	1	R
Votazione	Relazione	5	R

Table 3: Tavola degli accessi dell'operazione Visualizzazione scheda contenuto **senza ridondanza**

Concetto	Costrutto	Accessi	Tipo
Votazione	Relazione	1	W

Table 4: Tavola degli accessi dell'operazione votazione contenuto **senza ridondanza**

R: Read **W:** Write

Dal punto di vista della memoria l'attributo Voto in contenuto dovendo contenere un numero float (real) compreso tra 1 e 5, occuperà 4 bytes di spazio, che sono decisamente trascurabili. ($4 \cdot 100000 = 0,4\text{Megabytes}$)

Nello scenario con la ridondanza si ha l'operazione di Visualizzazione contenuto trascurabile, mentre l'operazione di votazione più pesante da mantenere in quanto richiede la lettura delle precedenti votazioni per quello stesso contenuto.

D'altro canto senza ridondanza l'operazione più pesante diventa quella del calcolo della media voti, mentre l'operazione di votazione è trascurabile.

Considerando però che il numero di visualizzazioni giornaliere della scheda contenuti (5000) è nettamente superiore a quella delle votazioni (500) (oltre alla sottostima di soli 5 voti per contenuto, nella realtà sono sicuramente di più) si decide di mantenere la ridondanza, che verrà aggiornata tramite trigger in seguito a una votazione che avverrà con meno frequenza.

Accessi totali giornalieri con ridondanza:

$$\begin{aligned}\text{Visualizzazione scheda contenuto:} & 5000 \cdot 1 = 5000 \\ \text{Votazione:} & 500 \cdot (4 + (1 \cdot 2) + (1 \cdot 2)) = 4000 \\ \text{Totale:} & 5000 + 4000 = 9000\end{aligned}$$

Accessi totali giornalieri senza ridondanza:

$$\begin{aligned}\text{Visualizzazione scheda contenuto:} & 5000 \cdot 6 = 30000 \\ \text{Votazione:} & 500 \cdot (1 \cdot 2) = 1000 \\ \text{Totale:} & 30000 + 1000 = 31000 \gg 9000\end{aligned}$$

Partecipazione / Partecipazione speciale

Un'altra ridondanza è rappresentata dall'associazione Partecipazione (reificata in entità) su Contenuto e Partecipazione speciale (reificata in entità) su Episodio.

Episodio essendo collegato alla entità Serie, eredita le associazioni di Contenuto.

In questo caso possiamo esplicitare una funzione diversa per le 2 associazioni che permettono di preservare la copertura delle funzionalità richieste e nel contempo di eliminare la ridondanza.

In particolare, possiamo considerare l'associazione Partecipazione tra Contenuto e Personaggio come la definizione del Cast di default, mentre l'associazione Partecipazione speciale tra Episodio e Personaggio può essere vista come un override del cast, che contiene soltanto le eccezioni per un determinato episodio.

In questo modo non sarà necessario specificare per ogni episodio tutto il cast, dato che verrà desunto dalla serie di appartenenza. Tale operazione sarà necessaria in caso di eccezioni, con un considerevole risparmio di inserimento di tuple.

Attributo Anno

Una ulteriore ridondanza è rappresentata dall'attributo Anno in Contenuto. Infatti l'anno di pubblicazione di un contenuto può essere desunto ed estratto tramite funzioni sql dalla data di pubblicazione. Data la semplicità di estrazione e la mancanza di valore aggiunto si decide di rimuovere la ridondanza.

Attributo Età

La medesima considerazione si può fare per l'attributo Età in Personaggio dato che può essere agevolmente calcolata a partire dalla data odierna. Data la semplicità di estrazione e la mancanza di valore aggiunto si decide di rimuovere la ridondanza.

2.3.2 Eliminazione delle generalizzazioni

Le generalizzazioni presenti nell'attuale schema ER sono: Utente (parziale e disgiunta) con un sottoinsieme Redattore e Contenuto (totale e disgiunta) formata dai sottoinsiemi Film, Serie TV e Programma.

2.3.3 Partizionamento/Accorpamento di entità e associazioni

In entrambi i casi si decide di tradurre le generalizzazioni con l'accorpamento delle entità figlie nel padre.

Gli svantaggi principali dati da questo design consistono principalmente nello spreco di memoria, dato che gli attributi specifici delle entità figlie non vengono valorizzati quando l'occorrenza è di tipo diverso, risultando in valori null.

In entrambi i casi lo spreco di memoria è irrisorio dato che gli attributi aggiuntivi sono rispettivamente:

- Data inizio e fine collaborazione per Redattore
- Un unico attributo *Tipo*, atto a determinare la tipologia di ogni istanza di Contenuto.

A tal proposito, l'attributo Tipo è sufficiente, in quanto la generalizzazione è disgiunta. Non sono quindi contemplati casi in cui un Film è anche Programma Tv o una Serie è anche un Film, ecc.

Tale soluzione garantisce inoltre minori accessi e quindi migliori prestazioni per funzionalità (operazioni) come ricerca di contenuti senza specificarne il tipo.

Nel caso degli utenti invece, non è necessario aggiungere l'attributo tipo per identificare i redattori, in quanto l'informazione può essere desunta controllando il valore dell'attributo Data inizio collaborazione.

Per sopperire al caso in cui un redattore interrompa le sue prestazioni di lavoro, introduciamo un ulteriore attributo *Data fine collaborazione* che permette di non dover cancellare le informazioni relative all'utenza e di mantenere lo storico delle sue redazioni di schede contenuto.

Con tale modifica, per determinare se un utente è un redattore piuttosto che un visitatore, sarà sufficiente verificare che data inizio collaborazione abbia un valore (passato) e che data fine collaborazione non abbia valore (o valore futuro).

2.3.4 Scelta degli identificatori principali

Per poter risolvere le inefficienze legate all'utilizzo di chiavi composte, chiavi alfanumeriche particolarmente lunghe o chiavi con identificatori esterni, sono stati introdotti i seguenti identificatori principali:

- Id per l'associazione Votazione (supporto cancellazione utente)
- Id per l'associazione Redazione (supporto cancellazione utente)
- Id per l'entità Sala
- Id per l'entità Cinema
- Id per l'entità Episodio
- Id per l'entità Personaggio
- Id per l'entità Contenuto
- Id per l'entità Piattaforma

- Id per l'entità Partecipazione
- Id per l'entità Partecipazione speciale

Si decide di mantenere Username come chiave di Utente in quanto si ipotizza di creare delle regole di business che ne vincolino la composizione come caratteri consentiti e lunghezza minima / massima.

2.4 Schema E-R ristrutturato + business rules

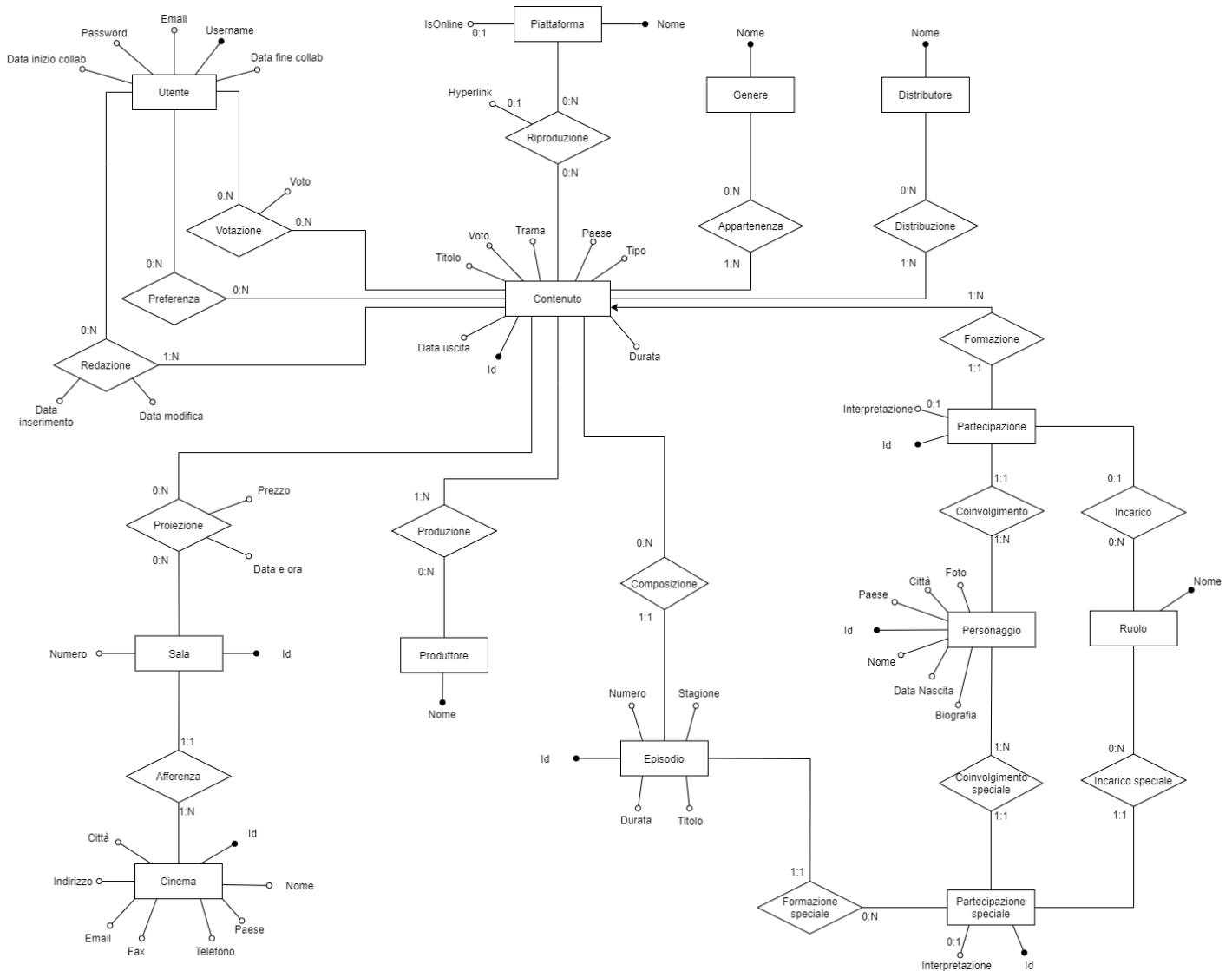


Figure 3: Schema ER Ristrutturato

Le business rules sono le medesime già definite nel capitolo 1 più la seguente:
Regole di vincolo

- Un contenuto può essere proiettato in un cinema soltanto se è di tipo Film.

2.5 Schema relazionale

Lo schema ER ristrutturato viene tradotto nelle seguenti relazioni:

utente(username, email, password, data_inizio_collaborazione, data_fine_collaborazione)
contenuto(id, titolo, voto, trama, paese, data_uscita, tipo, durata)
partecipazione(id, id_contenuto, id_personaggio, ruolo, interpretazione)
partecipazione_speciale(id, id_episodio, id_personaggio, ruolo, interpretazione)
ruolo(nome)
piattaforma(nome, is_online)
genere(nome)
distributore(nome)
personaggio(id, biografia, data_nascita, nome, paese, città, foto)
episodio(id, serie, numero, stagione, durata, titolo)
produttore(nome)
sala(id, numero, id_cinema)
cinema(id, nome, paese, email, telefono, fax, indirizzo, città)
contenuto_votazione(id, id_contenuto, username, voto)
contenuto_preferenza(id, id_contenuto, username)
contenuto_redazione(id, id_contenuto, username, data_inserimento, data_modifica)
contenuto_riproduzione(id, id_contenuto, piattaforma, hyperlink)
contenuto_genere(id, id_contenuto, genere)
contenuto_distribuzione(id, id_contenuto, distributore)
contenuto_produzione(id, id_contenuto, produttore)
contenuto_proiezione(id, id_contenuto, id_sala, data_ora, prezzo)

Vincoli di integrità referenziale (in "dot notation"):

- id_contenuto in partecipazione referencia contenuto.id;
- id_personaggio in partecipazione referencia personaggio.id;
- ruolo in partecipazione referencia ruolo.nome;
- id_episodio in partecipazione_speciale referencia episodio.id;
- id_personaggio in partecipazione_speciale referencia personaggio.id;
- ruolo in partecipazione_speciale referencia ruolo.nome;
- id_cinema in sala referencia cinema.id;
- id_contenuto in contenuto_votazione referencia contenuto.id;
- username in contenuto_votazione referencia utente.username;
- id_contenuto in contenuto_preferenza referencia contenuto.id;
- username in contenuto_preferenza referencia utente.username;
- id_contenuto in contenuto_redazione referencia contenuto.id;

- username in contenuto_redazione referencia utente.username;
- id_contenuto in contenuto_riproduzione referencia contenuto.id;
- piattaforma in contenuto_riproduzione referencia piattaforma.nome;
- id_contenuto in contenuto_genere referencia contenuto.id;
- genere in contenuto_genere referencia genere.nome;
- id_contenuto in contenuto_distribuzione referencia contenuto.id;
- distributore in contenuto_distribuzione referencia distributore.nome;
- id_contenuto in contenuto_produzione referencia contenuto.id;
- produttore in contenuto_produzione referencia produttore.nome;
- id_contenuto in contenuto_proiezione referencia contenuto.id;
- id_sala in contenuto_proiezione referencia sala.id;

Si decide di adottare come naming convention (per omogeneità con l'implementazione sql) per relazioni (tabelle) e attributi *lowercase separato da underscore* in riferimento al fatto che, citando la documentazione ufficiale "Key words and unquoted identifiers are case insensitive."

Si decide inoltre di denominare le relazioni risultanti da associazioni molti a molti con la seguente struttura: nome entità soggetto_nome associazione, quando il nome dell'associazione è autoesplicativo e sufficiente a interpretare la relazione intuitivamente.

Nel caso particolare di Contenuto - Appartenenza - Genere, si decide invece di adottare contenuto_genere.

La scelta di anteporre il nome dell'entità soggetto è voluta, per semplificare la ricerca e l'uso pratico della base dati.

source: <https://www.postgresql.org/docs/current/sql-syntax-lexical.html#SQL-SYNTAX-IDENTIFIERS>

3 Implementazione

L'ambiente di sviluppo viene creato tramite un container docker tramite docker-compose di cui si allega il file descrittore :

```
1 version: '3.1'
2
3 services:
4
5   db:
6     image: postgres
7     ports:
8       - 5432:5432
9     env_file:
10      - database.env # postgres configuration file
                        containing login information
11    volumes:
12      - ./schema.sql:/docker-entrypoint-initdb.d/1-
        schema.sql # execs ddl script at startup
13      - ./data.sql:/docker-entrypoint-initdb.d/2-data.
        sql # execs dml script at startup
14
15 volumes:
16   database-data:
```

3.1 DDL: Creazione del Database

```
1 BEGIN;
2
3 -- CREATE DATABASE comingsoon;
4
5 CREATE TYPE content_type AS ENUM ('film', 'serie TV', '
  programma');
6
7 CREATE TABLE utente (
8   username varchar(50) NOT NULL,
9   nome varchar(100) NOT NULL,
10  email varchar(100) NOT NULL,
11  password varchar(50) NOT NULL,
12  data_inizio_collaborazione timestamp NULL,
13  data_fine_collaborazione timestamp NULL
14      CHECK (data_inizio_collaborazione IS NULL OR
        data_fine_collaborazione IS NULL OR
        data_fine_collaborazione >
        data_inizio_collaborazione),
15  CONSTRAINT pk_utente PRIMARY KEY (username)
16 );
17
18 CREATE TABLE contenuto(
19   id int NOT NULL GENERATED BY DEFAULT AS IDENTITY,
20   titolo varchar(500) NOT NULL,
21   voto real DEFAULT 0 NOT NULL CHECK (voto >= 0 AND voto
        <= 5),
22   trama text NOT NULL,
23   paese varchar(50) NOT NULL,
24   data_uscita date NOT NULL,
25   tipo content_type NOT NULL,
26   durata smallint NULL,
```

```

27     CONSTRAINT pk_contenuto PRIMARY KEY (id)
28 );
29
30 CREATE TABLE ruolo(
31     nome varchar(50) NOT NULL,
32     CONSTRAINT pk_ruolo PRIMARY KEY (nome)
33 );
34
35 CREATE TABLE personaggio(
36     id int NOT NULL GENERATED BY DEFAULT AS IDENTITY,
37     biografia varchar(10000) NULL,
38     data_nascita date NOT NULL,
39     nome varchar(100) NOT NULL,
40     paese varchar(100) NOT NULL,
41     citta varchar(100) NOT NULL,
42     foto bytea NULL,
43     CONSTRAINT pk_personaggio PRIMARY KEY (id)
44 );
45
46 CREATE TABLE episodio(
47     id int NOT NULL GENERATED BY DEFAULT AS IDENTITY,
48     serie int NOT NULL REFERENCES contenuto,
49     numero int NOT NULL,
50     stagione int NOT NULL,
51     durata int NOT NULL,
52     titolo varchar(500) NOT NULL,
53     CONSTRAINT pk_episodio PRIMARY KEY (id),
54     UNIQUE (numero, stagione, serie)
55 );
56
57 CREATE TABLE partecipazione(
58     id int NOT NULL GENERATED BY DEFAULT AS IDENTITY,
59     id_contenuto int NOT NULL REFERENCES contenuto,
60     id_personaggio int NOT NULL REFERENCES personaggio,
61     ruolo varchar(50) NOT NULL REFERENCES ruolo,
62     interpretazione varchar(100) NULL
63     CHECK(interpretazione IS NULL OR ruolo = 'attore')
64     ,
65     CONSTRAINT pk_partecipazione PRIMARY KEY (id),
66     UNIQUE (id_contenuto, id_personaggio, ruolo,
67             interpretazione)
68 );
69
70 CREATE TABLE partecipazione_speciale(
71     id int NOT NULL GENERATED BY DEFAULT AS IDENTITY,
72     id_episodio int NOT NULL REFERENCES episodio,
73     id_personaggio int NOT NULL REFERENCES personaggio,
74     ruolo varchar(50) NOT NULL REFERENCES ruolo,
75     interpretazione varchar(100) NULL,
76     CONSTRAINT pk_partecipazione_speciale PRIMARY KEY (id)
77     ,
78     UNIQUE (id_episodio, id_personaggio, ruolo,
79             interpretazione)
80 );
81
82 CREATE TABLE piattaforma(
83     nome varchar(50) NOT NULL,
84     is_online bit NOT NULL,

```



```

81     CONSTRAINT pk_piattaforma PRIMARY KEY (nome)
82 );
83
84 CREATE TABLE genere(
85     nome varchar(50) NOT NULL,
86     CONSTRAINT pk_genere PRIMARY KEY (nome)
87 );
88
89 CREATE TABLE distributore(
90     nome varchar(100) NOT NULL,
91     CONSTRAINT pk_distributore PRIMARY KEY (nome)
92 );
93
94 CREATE TABLE produttore(
95     nome varchar(100) NOT NULL,
96     CONSTRAINT pk_prouttore PRIMARY KEY (nome)
97 );
98
99 CREATE TABLE cinema(
100     id int NOT NULL GENERATED BY DEFAULT AS IDENTITY,
101     nome varchar(150) NOT NULL,
102     indirizzo varchar(150) NOT NULL,
103     paese varchar(100) NOT NULL,
104     citta varchar(100) NOT NULL,
105     email varchar(150) NULL,
106     telefono varchar(20) NULL,
107     fax varchar(20) NULL,
108     CONSTRAINT pk_cinema PRIMARY KEY (id)
109 );
110
111 CREATE TABLE sala(
112     id int NOT NULL GENERATED BY DEFAULT AS IDENTITY,
113     numero int NOT NULL,
114     id_cinema int NOT NULL REFERENCES cinema ON DELETE
115         CASCADE,
116     CONSTRAINT pk_Sala PRIMARY KEY (id)
117 );
118
119 CREATE TABLE contenuto_votazione(
120     id int NOT NULL GENERATED BY DEFAULT AS IDENTITY,
121     id_contenuto int NOT NULL REFERENCES contenuto ON
122         DELETE CASCADE,
123     username varchar(50) NULL REFERENCES utente ON DELETE
124         SET NULL,
125     voto smallint NOT NULL CHECK (voto >= 0 AND voto <= 5)
126     ,
127     CONSTRAINT pk_contenuto_votazione PRIMARY KEY (id),
128     UNIQUE (id_contenuto, username)
129 );
130
131 CREATE TABLE contenuto_preferenza(
132     id_contenuto int NOT NULL REFERENCES contenuto ON
133         DELETE CASCADE,
134     username varchar(50) NOT NULL REFERENCES utente ON
135         DELETE CASCADE,
136     CONSTRAINT pk_contenuto_preferenza PRIMARY KEY (
137         id_contenuto, username)
138 );

```

```

132
133 CREATE TABLE contenuto_redazione(
134     id int NOT NULL GENERATED BY DEFAULT AS IDENTITY,
135     id_contenuto int NOT NULL REFERENCES contenuto ON
        DELETE CASCADE,
136     username varchar(50) NOT NULL REFERENCES utente ON
        DELETE SET NULL,
137     data_inserimento timestamp NOT NULL,
138     data_modifica timestamp NULL CHECK(data_modifica is
        NULL OR data_modifica > data_inserimento),
139     CONSTRAINT pk_contenuto_redazione PRIMARY KEY (id)
140 );
141
142 CREATE TABLE contenuto_riproduzione(
143     id_contenuto int NOT NULL REFERENCES contenuto ON
        DELETE CASCADE,
144     piattaforma varchar(50) NOT NULL REFERENCES
        piattaforma ON DELETE CASCADE,
145     hyperlink varchar(500) NULL,
146     CONSTRAINT pk_contenuto_riproduzione PRIMARY KEY (
        id_contenuto, piattaforma)
147 );
148
149 CREATE TABLE contenuto_genere(
150     id_contenuto int NOT NULL REFERENCES contenuto ON
        DELETE CASCADE,
151     genere varchar(50) NOT NULL REFERENCES genere ON
        DELETE CASCADE,
152     CONSTRAINT pk_contenuto_genere PRIMARY KEY (
        id_contenuto, genere)
153 );
154
155 CREATE TABLE contenuto_distribuzione(
156     id_contenuto int NOT NULL REFERENCES contenuto ON
        DELETE CASCADE,
157     distributore varchar(100) NOT NULL REFERENCES
        distributore ON DELETE CASCADE,
158     CONSTRAINT pk_contenuto_distribuzione PRIMARY KEY (
        id_contenuto, distributore)
159 );
160
161 CREATE TABLE contenuto_produzione(
162     id_contenuto int NOT NULL REFERENCES contenuto ON
        DELETE CASCADE,
163     produttore varchar(100) NOT NULL REFERENCES produttore
        ON DELETE CASCADE,
164     CONSTRAINT pk_contenuto_produzione PRIMARY KEY (
        id_contenuto, produttore)
165 );
166
167 CREATE TABLE contenuto_proiezione(
168     id_contenuto int NOT NULL REFERENCES contenuto ON
        DELETE CASCADE,
169     id_sala int NOT NULL REFERENCES sala ON DELETE CASCADE
        ,
170     data_ora timestamp NOT NULL,
171     prezzo decimal(5, 2) NOT NULL,
172     CONSTRAINT pk_contenuto_proiezione PRIMARY KEY (

```

```

        id_contenuto, id_sala, data_ora)
173 );
174
175
176 CREATE OR REPLACE FUNCTION update_content_rating_avg()
177 RETURNS trigger AS
178 $BODY$
179 BEGIN
180     UPDATE contenuto
181     SET voto = (
182         select avg(voto)::real
183         from contenuto_votazione
184         where id_contenuto = NEW.id_contenuto
185     )
186     WHERE id = NEW.id_contenuto;
187
188     RETURN NEW;
189 END;
190 $BODY$ LANGUAGE plpgsql;
191
192 CREATE TRIGGER update_rating
193 AFTER INSERT ON contenuto_votazione
194 FOR EACH ROW
195 EXECUTE PROCEDURE update_content_rating_avg();
196
197 COMMIT;

```

3.2 DML: Popolamento tabelle

```

1 INSERT INTO utente(
2     username,
3     nome,
4     email,
5     password,
6     data_inizio_collaborazione,
7     data_fine_collaborazione
8 )
9 VALUES (
10     'samuele_pecetto',
11     'Samuele Pecetto',
12     'samuele.pecetto@edu.unito.it',
13     'password',
14     '2020-07-08',
15     NULL
16 ),
17 (
18     'john_doe',
19     'John Doe',
20     'john.doe@visitor.com',
21     'password',
22     NULL,
23     NULL
24 ),
25 (
26     'rose_tan',
27     'Rose R. Tan',
28     'email@visitor.com',
29     'password',
30     NULL,

```

```

31         NULL
32     ),
33     (
34         'michael_white',
35         'Michael P. White',
36         'email@visitor.com',
37         'password',
38         NULL,
39         NULL
40     ),
41     (
42         'daniel_tay',
43         'Daniel J. Tay',
44         'email@visitor.com',
45         'password',
46         '2020-07-21 8:45:00',
47         NULL
48     ),
49     (
50         'donna_nelson',
51         'Donna J. Nelson',
52         'email@visitor.com',
53         'password',
54         NULL,
55         NULL
56     ),
57     (
58         'judy_lawson',
59         'Judy R. Lawson',
60         'email@visitor.com',
61         'password',
62         NULL,
63         NULL
64     ),
65     (
66         'johnatan_crepeau',
67         'Jonathan L. Crepeau',
68         'email@visitor.com',
69         'password',
70         '2018-04-01 12:15:00',
71         '2020-05-20 18:35:00'
72     ),
73     (
74         'diana_mclaughlin',
75         'Diana W. McLaughlin',
76         'email@visitor.com',
77         'password',
78         NULL,
79         NULL
80     ),
81     (
82         'mark_mertz',
83         'Mark J. Mertz',
84         'email@visitor.com',
85         'password',
86         NULL,
87         NULL
88     );

```

```

89  INSERT INTO contenuto(
90      id,
91      titolo,
92      trama,
93      paese,
94      data_uscita,
95      tipo,
96      durata
97  )
98  VALUES (
99      1,
100     'You Can Count on Me',
101     'A single mother''s life is thrown into turmoil
        after her struggling, rarely-seen younger
        brother returns to town.',
102     'USA',
103     '2000-12-22',
104     'film',
105     111
106 ),
107 (
108     2,
109     'Come diventare grandi nonostante i genitori',
110     'Sempre piu'' spesso i genitori assumono
        comportamenti competitivi verso i professori
        dei propri figli: contestano voti e programmi,
        vaneggiano di simpatie, antipatie e complotti.
        Così'', invece di aiutarli nella formazione dei
        loro ragazzi, diventano ostacoli
        insormontabili alla loro crescita.
        Presuntuosamente pensano: "Noi conosciamo
        meglio di chiunque altro i nostri figli e
        sappiamo quanto valgono e come e cosa gli si
        deve insegnare". E'' quello che accade anche ai
        ragazzi di "Come diventare grandi nonostante i
        genitori" quando, al liceo, arriva la nuova
        preside che decide di non aderire al concorso
        scolastico nazionale per gruppi musicali. Per i
        ragazzi, che hanno una passione sfrenata per
        la musica, e'' un colpo mortale e, anche quando
        i genitori corrono a protestare, la preside
        decide addirittura di raddoppiare il lavoro
        quotidiano dei ragazzi. Dopo i primi voti bassi
        , i genitori consigliano prudentemente ai
        ragazzi di sottostare alle decisioni della
        nuova preside. Tuttavia i ragazzi, con orgoglio
        , decidono di iscriversi al concorso musicale
        pur avendo contro scuola e genitori. La sfida
        pare impossibile e invece porterà'' i ragazzi a
        crescere in modo sorprendente tra ostacoli di
        ogni tipo da superare.',
111     'Italia',
112     '2016-11-24',
113     'film',
114     90
115 ),
116 (
117     3,

```

118 'Un sacco bello',
 119 'Un sacco bello e'' un film scritto e diretto
 Carlo Verdone, il quale interpreta i tre
 protagonisti e alcuni personaggi secondari.
 120 In una Roma semi deserta, sotto al sole cocente di agosto,
 si svolgono le vicende dei tre personaggi principali.
 Enzo e'' un giovanotto spavaldo che ama raccontare le
 sue incredibili avventure con le donne. Intenzionato ad
 arrivare con la sua fiammante automobile sportiva fino
 a Cracovia, e'' in cerca di un compagno di viaggio.
 Dopo una lunga ricerca, riesce a convincere l''amico
 Sergio (Renato Scarpa), che pero'' ha un malore appena
 usciti dalla capitale. Così'' Enzo deve trovare al piu'
 ' presto un rimpiazzo.
 121 L''impacciato Leo vive a Trastevere con la madre, che lo
 aspetta al mare per passare Ferragosto assieme. Il
 ragazzo viene pero'' bloccato da Marisol (Veronica
 Miriel), una bella turista spagnola che gli chiede
 aiuto. Timido e ingenuo, Leo si trova costretto ad
 ospitare in casa l''esuberante straniera. Ma proprio
 quando tra i due sembra nascere del tenero, ecco che
 arriva il fidanzato di lei...
 122 Dopo una lunga permanenza in una comunita'' di hippie,
 Ruggero e la sua ragazza Fiorenza (Isabella De Bernardi
) arrivano a Roma. Lì'' il ragazzo rincontra il padre (
 Mario Brega), che li invita a casa. L''uomo spera di
 convincere il figlio a ritornare sulla retta via,
 contando sull''aiuto di padre Alfio e del professor
 Anselmo.',
 123 'Italia',
 124 '1980-01-01',
 125 'film',
 126 99
 127),
 128 (
 129 4,
 130 'Quarto potere',
 131 'Quarto potere, e'' un film del 1941 scritto,
 diretto ed interpretato da Orson Welles. Il
 coinvolgente lungometraggio narra la vita del
 famoso magnate della stampa Charles Foster Kane
 (Orson Welles), uomo dalla personalita''
 complessa, incapace di amare gratuitamente gli
 altri e condannato per questo a vivere una
 lunga parte della sua vita in solitudine nel
 Castello di Candalu'', in Florida, fino al
 momento della sua dipartita.',
 132 'USA',
 133 '1941-01-01',
 134 'film',
 135 120
 136),
 137 (
 138 5,
 139 'Tempi moderni',
 140 'Tempi moderni, e'' un film muto del 1936, diretto
 da Charlie Chaplin. Charlot (Charlie Chaplin)
 lavora come operaio in una fabbrica industriale

e la sua mansione e'' quella di stringere bulloni. Quando viene scelto per un esperimento, in cui una macchina potra'' nutrirlo mentre continua a lavorare (evitando cosi'' la perdita di tempo della pausa pranzo), l''uomo, gia'' stressato dal gesto ripetuto meccanicamente all''infinito, impazzisce e ormai fuori di senno, manomette tutti i macchinari della fabbrica, finche'' viene portato via e ricoverato in una clinica per essere riabilitato.',

141 'USA',
142 '1936-01-01',
143 'film',
144 85
145),
146 (
147 6,
148 'Psycho',
149 'Da quasi quarantanni Psycho e'' sinonimo di
suspance cinematografica. Psycho si e''
insinuato nell'' anima di tutti coloro che si
sono avventurati nella realta'' del Motel bates
fosca, ma indubbiamente seducente e dei suoi
insoliti gestori: Norman Bates e sua madre, una
donna anziana e autoritaria. Psycho rimane,
sotto ogni punto di vista, un capolavoro. Al
centro della vicenda, la provocante Marion
Crane, una giovane donna che, nel disperato
tentativo di cambiare vita, agendo d''impulso
deruba il suo datore di lavoro. Con le valigie
pronte, il denaro rubato nella borsetta e nella
speranza di un nuovo inizio di fronte a lei, la
donna si affretta a lasciare la citta''.
Quando si avvicina la notte, la giovane si
rifugia in un motel fuori mano. Non del tutto
pura, ne'' del tutto cattiva, Marion cade
vittima della psicosi per il solo motivo di
trovarsi nel posto sbagliato al momento
sbagliato. Allo stesso modo Norman Bates, il
proprietario dalle maniere gentili, era
lantitesi di ogni convenzione tipica dei
personaggi cinematografici. Di bell''aspetto,
tranquillo e appena un po strano, Norman si
rivelava essere il relitto di un uomo afflitto
dal complesso di Edipo, con un''inclinazione a
fare cose sconvenienti con i coltelli da cucina
.',

150 'USA',
151 '1999-04-23',
152 'film',
153 103
154),
155 (
156 7,
157 'Breaking Bad',
158 'Walter White, sottopagato insegnante di chimica
alla soglia dei cinquanta anni, scopre di avere
un cancro terminale ai polmoni. Incapace di

```

accettare questa terribile realta'', e
preoccupato sul futuro della sua famiglia - una
moglie incinta e un figlio affetto da paralisi
celebrale - Walter decide di tirare fuori la
rabbia e sfruttare le sue conoscenze di chimica
per produrre metamfetamina insieme a Jesse, un
suo ex studente nel giro gia'' da diverso
tempo.',
159      'USA',
160      '2008-01-01',
161      'serie TV',
162      NULL
163  ),
164  (
165      8,
166      'Game of Thrones',
167      'Il Trono di Spade, la Serie TV creata da David
      Benioff e D.B. Weiss e tratta dai romanzi di
      George R. R. Martin, vede sette nobili famiglie
      che lottano per il controllo della mitologica
      terra di Westeros. Robert Baratheon, re di
      Westeros, chiede al suo vecchio amico Eddard
      Stark di servirlo come suo principale
      consigliere. Eddard accetta, ma ha dei sospetti
      . Mentre la famiglia del re, i Lannister, cova
      un complotto per assumere il controllo del
      trono, dall''altra parte del mare, i Targaryen
      si preparano per un ritorno al potere. Il
      conflitto tra queste due famiglie e altre
      porteranno alla guerra, mentre a nord un''
      antica minaccia si risveglia.',
168      'USA',
169      '2011-01-01',
170      'serie TV',
171      NULL
172  ),
173  (
174      9,
175      'Babylon Berlin',
176      'Babylon Berlin e'' una serie tv drammatica
      tedesca che narra la storia del Commissario
      Gereon Rath (Volker Bruch, The Reader)
      trasferito per una missione segreta da Colonia
      a Berlino per indagare su un giro di estorsioni
      che sembrano coinvolgere anche il sindaco.
      Nelle sue indagini il commissario Rath e''
      coadiuvato dal corpo di polizia della squadra
      del Forte Rosso la gigantesca sede della
      polizia di Berlino, guidata da Bruno Wolter (
      Peter Kurth, Gold) un uomo attento e meticoloso
      nel proprio lavoro ma che nasconde un secondo
      fine pericoloso. ',
177      'Germania',
178      '2017-01-01',
179      'serie TV',
180      NULL
181  ),
182  (

```


183 10,
184 'Antonino Chef Academy',
185 'Antonino Chef Academy e'' un programma culinario
condotto dallo chef partenopeo Cannavacciuolo.
Nella location del Castello Dal Pozzo di
Oleggio, il cuoco pluripremiato deve vestire i
panni di professore, per guidare dieci giovani
aspiranti cuochi (tra i 18 e i 23 anni) che, a
seguito di un percorso formativo, si sfidano in
delle prove in cui devono mettere a frutto
tutte le loro conoscenze ai fornelli. ',
186 'Italia',
187 '2019-01-01',
188 'programma',
189 90
190),
191 (
192 11,
193 'Forum',
194 'l format della trasmissione e'' sempre rimasto
invariato: due persone in contrasto su un
argomento - problemi familiari, incidenti
stradali, liti di condominio e via dicendo - si
affidano alla sentenza di un giudice. Le
questioni affrontate sono ricostruzioni di
fatti, interpretate da due attori: il giudizio
finale non e'' quindi da ritenersi ufficiale.
Per ogni causa affrontata, le parti interessate
hanno diritto di esprimere il proprio punto di
vista. Al termine del dibattito tra i
contendenti, il giudice si ritira per prendere
una decisione e la presentatrice chiede un
parere al pubblico presente in studio.',
195 'Italia',
196 '1985-01-01',
197 'programma',
198 120
199),
200 (
201 12,
202 'Paperissima Sprint',
203 'Paperissima Sprint e'' un programma ideato da
Antonio Ricci, nato come spin-off estivo del
varietà'' Paperissima nel 1990 e attualmente in
onda la domenica in fascia preserale su
Canale5. Durante ogni puntata si alternano gli
sketch dei presentatori - affiancati dal
Gabibbo e dalle veline - a video divertenti
provenienti da tutto il globo: filmati di
piccoli incidenti con bambini, animali,
matrimoni, imprese sportive, il tutto
enfaticizzato da suoni simpatici che accompagnano
le scene. Spesso sono trasmesse anche gaffe di
vip durante delle riprese cinematografiche o
televisive. Dal 2005 il programma e'' trasmesso
anche in location esterne e durante la
stagione invernale. Nel corso delle edizioni si
sono susseguiti molti personaggi al timone

```

        della trasmissione, tra cui il celebre Mike
        Bongiorno, una giovane Michelle Hunziker, l''
        attrice Serena Grandi, la showgirl e''va Henger
        , la modella Juliana Moreira, la soubrette
        Giorgia Palmas, lo sportivo Vittorio Brumotti,
        la comica Valeria Graci, Maddalena Corvaglia e
        per ultima Roberta Lanfranchi.',
204     'Italia',
205     '1990-01-01',
206     'programma',
207     50
208 ),
209 (
210     13,
211     'Pulp Fiction',
212     'Pulp Fiction, film del 1994 scritto e diretto da
        Quentin Tarantino, e'' una storia composta da
        tre racconti distinti, in ordine non
        cronologico, che si sviluppano intrecciandosi
        in una sorta di percorso circolare, con inizio
        e fine al mattino e nello stesso posto, una
        caffetteria di Los Angeles chiamata Hawthorne
        Grill...',
213     'USA',
214     '1994-12-16',
215     'film',
216     153
217 );
218 INSERT INTO episodio(id, serie, numero, stagione, durata,
        titolo)
219 VALUES (
220     1,
221     7,
222     1,
223     1,
224     47,
225     'Questione di chimica'
226 ),
227 (
228     2,
229     7,
230     2,
231     1,
232     47,
233     'Senza ritorno'
234 ),
235 (
236     3,
237     7,
238     3,
239     1,
240     47,
241     'Conseguenze radicali'
242 ),
243 (
244     4,
245     7,
246     4,

```

```

247         1,
248         47,
249         'Una malattia scomoda'
250     ),
251     (
252         5,
253         7,
254         1,
255         2,
256         47,
257         'Tutto cambia'
258     );
259 INSERT INTO genere(nome)
260 VALUES ('commedia'),
261         ('comico'),
262         ('horror'),
263         ('thriller'),
264         ('drammatico'),
265         ('romantico'),
266         ('noir'),
267         ('giallo'),
268         ('western');
269 INSERT INTO cinema(
270     id,
271     nome,
272     indirizzo,
273     paese,
274     citta,
275     email,
276     telefono,
277     fax
278 )
279 VALUES (
280     1,
281     'Multisala Reposi',
282     'via XX settembre 15',
283     'Italia',
284     'Torino',
285     null,
286     '011 531400',
287     null
288 ),
289 (
290     2,
291     'Cinema Massimo',
292     'via verdi 18',
293     'Italia',
294     'Torino',
295     null,
296     '011 813 8574',
297     null
298 );
299 INSERT INTO sala(id, numero, id_cinema)
300 VALUES (1, 1, 1),
301         (2, 2, 1),
302         (3, 3, 1),
303         (4, 4, 1),
304         (5, 1, 2);

```

```

305 INSERT INTO piattaforma(nome, is_online)
306 VALUES ('netflix', 1::bit),
307         ('mediaset', 0::bit),
308         ('rai', 0::bit),
309         ('infinity', 1::bit);
310 INSERT INTO distributore(nome)
311 VALUES ('Medusa - Pentavideo'),
312         ('Medusa Video (Pepite)'),
313         ('Cecchi Gori Distribuzione');
314 INSERT INTO produttore(nome)
315 VALUES ('Sergio Leone'),
316         ('A Band Apart'),
317         ('Jersey Films'),
318         ('Miramax Films');
319 INSERT INTO ruolo(nome)
320 VALUES ('attore'),
321         ('regista'),
322         ('sceneggiatore'),
323         ('direttore fotografia'),
324         ('produttore musicale');
325 INSERT INTO personaggio(
326     id,
327     biografia,
328     data_nascita,
329     nome,
330     paese,
331     citta,
332     foto
333 )
334 VALUES (
335     1,
336     'Carlo Verdone, noto all''anagrafe anche con il
        secondo nome di Gregorio, e'' nato a Roma il 17
        novembre 1950 da Rossana Schiavina e dal
        critico cinematografico Mario Verdone, che ha
        contribuito in primis ad avvicinare il figlio
        al mondo del cinema, condividendo con lui la
        sua passione. Dopo la maturita'' classica e il
        diploma in regia al Centro Sperimentale di
        cinematografia di Roma, Verdone si e'' laureato
        con il massimo dei voti in Lettere Moderne con
        una tesi interdisciplinare in Letteratura e
        cinema muto italiano.',
337     '1950-11-17',
338     'Carlo Verdone',
339     'Italia',
340     'Roma',
341     NULL
342 ),
343 (
344     2,
345     'Attore. Nasce nel 1923 a ROMA (Italia). Muore a
        ROMA (Italia).',
346     '1923-03-05',
347     'Mario Brega',
348     'Italia',
349     'Roma',
350     NULL

```

```

351     ),
352     (
353         3,
354         NULL,
355         '1963-07-12',
356         'Isabella De Bernardi',
357         'Italia',
358         'Roma',
359         NULL
360     ),
361     (
362         4,
363         'Compositore, direttore d''orchestra, autore di
            canzoni e arrangiatore. Nato a Roma nel 1928 e
            diplomato in tromba, composizione e direzione d
            ''orchestra al Conservatorio di Santa Cecilia
            col maestro Goffredo Petrassi, negli anni
            Cinquanta inizia a lavorare come suonatore di
            tromba in varie jazz band, prima di approdare
            al cinema componendo canzoni...',
364         '1928-11-10',
365         'Ennio Morricone',
366         'Italia',
367         'Roma',
368         NULL
369     ),
370     (
371         5,
372         'Quentin Jerome Tarantino nasce nel 1963 a
            Knoxville, nel Tennessee, da padre attore e da
            mamma Connie, che lo chiama Quentin in onore di
            Quint Asper, il personaggio di Burt Reynolds
            in Gunsmoke. Quando ha tre anni si trasferisce
            con la mamma divorziata a Los Angeles e
            dintorni.',
373         '1963-03-27',
374         'Quentin Tarantino',
375         'USA',
376         'Knoxville, Tennessee',
377         NULL
378     ),
379     (
380         6,
381         'Samuel L. Jackson, il futuro "King of cool",
            cresce a Chattanooga, nel Tennessee. Il padre e
            '' assente e alcolizzato: ad allevarlo e'' la
            madre Elizabeth, operaia, coi nonni materni, e
            a farlo appassionare al cinema e'' la zia, che
            lo porta spesso ai doppi spettacoli dell''epoca
            ... ',
382         '1948-12-21',
383         'Samuel L. Jackson',
384         'USA',
385         'WASHINGTON D.C',
386         NULL
387     ),
388     (
389         7,

```

```

390         NULL,
391         '1954-12-19',
392         'Andrzej Sekula',
393         'Polonia',
394         'Wroclaw',
395         NULL
396     ),
397     (
398         8,
399         NULL,
400         '1956-03-07',
401         'Bryan Cranston',
402         'USA',
403         'CANOGA PARK, CALIFORNIA',
404         NULL
405     ),
406     (
407         9,
408         NULL,
409         '1958-04-26',
410         'Giancarlo Esposito',
411         'Danimarca',
412         'COPENHAGEN',
413         NULL
414     );
415 INSERT INTO contenuto_votazione(id_contenuto, username,
voto)
416 VALUES (3, 'john_doe', 5),
417         (1, 'john_doe', 5),
418         (1, 'samuele_pecetto', 2),
419         (3, 'samuele_pecetto', 5),
420         (3, 'rose_tan', 3),
421         (3, 'daniel_tay', 4),
422         (3, 'johnatan_crepeau', 1);
423 INSERT INTO partecipazione(
424     id_contenuto,
425     id_personaggio,
426     ruolo,
427     interpretazione
428 )
429 VALUES (3, 1, 'regista', NULL),
430     (
431         3,
432         1,
433         'sceneggiatore',
434         NULL
435     ),
436     (3, 1, 'attore', 'Leo'),
437     (3, 1, 'attore', 'Ruggero'),
438     (3, 1, 'attore', 'P.Alfio'),
439     (
440         3,
441         2,
442         'attore',
443         'Padre Di Ruggero'
444     ),
445     (
446         3,

```

```

447         3,
448         'attore',
449         'Fiorenza'
450     ),
451     (
452         3,
453         4,
454         'produttore musicale',
455         NULL
456     ),
457     (
458         7,
459         8,
460         'attore',
461         'Walter White'
462     );
463 INSERT INTO partecipazione_speciale(
464     id_episodio,
465     id_personaggio,
466     ruolo,
467     interpretazione
468 )
469 VALUES (
470     5,
471     9,
472     'attore',
473     'Gustavo ''Gus'' Fring'
474 );
475 INSERT INTO contenuto_redazione(
476     id_contenuto,
477     username,
478     data_inserimento,
479     data_modifica
480 )
481 VALUES (3, 'samuele_pecetto', '2020-07-21', NULL);
482 INSERT INTO contenuto_proiezione(id_contenuto, id_sala,
483     data_ora, prezzo)
484 VALUES (
485     13,
486     1,
487     '2020-07-23 19:00:00',
488     5.5
489 ),(
490     13,
491     2,
492     '2020-07-23 23:15:00',
493     5.0
494 );
495 INSERT INTO contenuto_genere(id_contenuto, genere)
496 VALUES (13, 'drammatico'),
497 (13, 'thriller'),
498 (3, 'commedia'),
499 (3, 'comico');
500 INSERT INTO contenuto_riproduzione(id_contenuto,
501     piattaforma, hyperlink)
502 VALUES (7, 'netflix', 'link_to_resource.uri');

```

3.3 Constraints check: Operazioni di cancellazione e modifica

```

1  -- CONSTRAINT CHECK PER data_modifica > data_inserimento
2  -- ERROR RAISED
3  UPDATE contenuto_redazione
4  SET data_modifica = '2020-07-20'
5  WHERE username = 'samuele_pecetto'
6  AND id_contenuto = 3;
7
8  -- CONSTRAINT CHECK PER 0 <= voto <= 5
9  -- ERROR RAISED
10 INSERT INTO contenuto_votazione(id_contenuto, username,
    voto)
11 VALUES (1, 'john_doe', 6);
12
13 -- CANCELLAZIONE UTENTE CHE HA VOTATO CONTENUTI -
    MANTENIMENTO MEDIA VOTI
14 -- NO ERRORS RAISED
15 DELETE FROM utente
16 WHERE username = 'john_doe';
17
18 -- ESTRAZIONE ANNO DI UN CONTENUTO
19 SELECT extract(year from data_uscita)
20 FROM contenuto;
21
22 -- AGGIORNAMENTO MEDIA VOTO TRAMITE TRIGGER
23 -- EXPECT UPDATED VOT IN CONTENUTO
24 INSERT INTO contenuto_votazione(id_contenuto, username,
    voto)
25 VALUES (3, 'diana_mclaughlin', 5);
26
27 -- CONSTRAINT CHECK PER INTERPRETAZIONE ESCLUSIVA DI
    ATTORE
28 -- ERROR RAISED
29 INSERT INTO partecipazione(
30     id_contenuto,
31     id_personaggio,
32     ruolo,
33     interpretazione
34 )
35 VALUES (3, 1, 'regista', NULL),
36 (
37     3,
38     1,
39     'sceneggiatore',
40     'personaggio di fantasia'
41 );

```