

Documentazione Progetto

Vincenzo Palma

Anno accademico 2022-2023



UNIVERSITÀ DEGLI STUDI
DI NAPOLI FEDERICO II

Indice

1	Progettazione Concettuale	3
1.1	Analisi dei Requisiti	3
1.2	Schema Concettuale	4
1.3	Dizionario	5
2	Ristrutturazione del modello concettuale	7
2.1	Analisi delle Ridondanze	7
2.2	Analisi delle Generalizzazioni	8
2.3	Eliminazione degli attributi multivalore	8
2.4	Eliminazione degli attributi strutturato	8
2.5	Analisi di Entità e Associazioni	8
2.6	Identificazione chiavi primarie	8
2.7	Schema Ristrutturato	9
3	Traduzione al Modello Logico	9

1 Progettazione Concettuale

1.1 Analisi dei Requisiti

Lo step di Analisi dei Requisiti consiste nell'individuare le informazioni rilevanti per poter soddisfare le richieste informative e funzionali del DB. È essenziale individuare all'interno del problema le *Entità*, le *Relazioni*, i *Vincoli*, le operazioni che gli utenti possono effettuare più frequentemente ed il numero di utenti che interagiranno contemporaneamente con il Database.

"Si assuma di dover progettare una base di dati per la gestione di diversi cinema multisala."

Consideriamo di dover gestire più cinema, avremo quindi un'entità *Cinema* per poter rappresentare il cinema del mondo reale. Essa avrà come attributi il nome del cinema, il luogo nella quale è situato ed il numero di telefono.

"Ogni sala ha caratteristiche specifiche per le apparecchiature tecniche (schermo, proiezione, impianto audio), per la capienza, per la dimensione."

Non c'è necessità di dover rappresentare come entità a se le apparecchiature, servono solo a scopo informativo, di conseguenza saranno *attributi dell'entità sala*. Altri attributi che inseriremo solo per avere informazioni aggiuntive sulla sala, saranno la sua dimensione e la capienza massima.

"Ciascuna poltrona di ogni sala deve essere individuata univocamente ed ha comunque indicazione di fila e posizione nella fila."

Si richiede di tener traccia delle poltrone mediante *fila* e la *posizione nella fila*, di conseguenza saranno dell'entità, però queste entità non esisteranno se esse non saranno associate a delle sale, motivo per la quale saranno *entità deboli*.

Il sistema deve gestire la programmazione dei film nel multisala. In particolare:

- *La proiezione di un film in una sala;*
- *La proiezione degli spot pubblicitari che precedono il film;*
- *La proiezione delle anteprime dei film che possono precedere i film;*

E' necessario gestire le singole proiezioni. Ogni proiezione è identificata dalla data, dall'ora di inizio e dall'ora di fine ed è in relazione con un certo numero di *spot*, di *anteprime* e con un solo *film*, che saranno come già detto entità a se stanti.

Il sistema deve gestire la interazione con i clienti registrati. In particolare:

- Avremo un'entità *Cliente Registrato* che identificherà un singolo utente, ed un'entità *vendita* utile per gestire le prenotazioni, che siano online o da botteghino.

Gestione dei feedback degli spettatori al film

- Tramite un'entità *Biglietto* teniamo traccia del suo prezzo, così da poter calcolare l'incasso totale di una proiezione. Mentre il feedback per il momento è una relazione tra il cliente ed il film.

Disegno concettuale del nostro database utile a dare una rappresentazione del nostro *Mini-Mondo*.



1.3 Dizionario

Classi	Descrizione	Attributi
Cinema	Singolo cinema.	Nome (String): Indica il nome del cinema; Indirizzo-Città (String):Luogo nella quale è situato il cinema; NumTelefono (String): Recapito telefonico per poter contattare la sede del cinema;
Sala	Una sala del cinema multisala.	Schermo (String): Tipologia di schermo presente nella sala; ImpiantoAudio (String): Tipologia di impianto audio presente nella sala; ImpiantoVideo (String): Tipologia di impianto video presente nella sala; Dimensione (Double): Dimensione della sala in m ² ; Capienza (Int): Numero massimo di posti a sedere;
Posto	Singolo posto a sedere di una sala del cinema.	Fila (Char): Identifica univocamente la fila nella quale si trova una poltrona; Numero (Int): Identifica univocamente la colonna nella quale si trova una poltrona;
Biglietto	Un biglietto che riserva uno specifico posto affiliato ad una proiezione.	Matrice (String): Identifica univocamente un singolo biglietto; DataEmissione (Date): Data nella quale il biglietto viene venduto; Prezzo (Double): Costo del singolo biglietto;
Prenotazione	Prenotazione realizzata da un cliente associata ad un biglietto.	
Vendita	Una vendita è collegata ad un biglietto esplicitando il metodo di acquisto, Nonchè si suddivide in VenditaOnline e VenditaBotteghino.	Costo (Double): Costo del biglietto più eventuali commissioni di vendita;

Classi	Descrizione	Attributi
Proiezione	Una proiezione è una collezione formata da un film, da più spot e da più anteprime, che saranno trasmesse in una sala.	Data (Date): Data della messa in onda di una proiezione; OraInizio (Time): Orario di inizio della proiezione; OraFine (Time): Orario di fine della proiezione;
Film	Un film trasmesso durante una proiezione in una sala.	Titolo (String): Titolo del film; Anno (Int): Anno di uscita del film; Regista (String): Nome e Cognome del regista del film; cast (Attore): Nome e Cognome ripetuti più volte per identificare gli attori che hanno partecipato al film;
Cliente Registrato	Un Cliente che si è registrato sulla piattaforma online.	Username (String): Identifica univocamente il cliente all'interno del DB; Password (String): Permette al Cliente di accedere al suo account online; DataRegistrazione (Date): Specifica la data di registrazione del cliente sulla piattaforma online;
Anteprima	Trailer di un film.	Durata (Int): Durata del trailer;
Spot	Video utile a sponsorizzare un prodotto, trasmesso prima e dopo il film.	Azienda (String): Nome dell'azienda che decide di sponsorizzare un prodotto; Durata (Int): Durata dello spot pubblicitario;

Associazioni	
Appartenenza	Associazione uno-a-molti che mette in relazione Uno o più posti in una sala.
Possesso	Associazione uno-a-molti che mette in relazione Uno o più Sale in un Cinema.
Riservo	Associazione uno-a-molti che mette in relazione un biglietto ad un posto a sedere nella sala.
Associazione	Associazione uno-a-uno che mette in relazione un biglietto ad una prenotazione.
Collegamento	Associazione uno-a-uno che mette in relazione un biglietto ad una vendita, che sia online o al botteghino.
Realizzazione	Associazione uno-a-molti che mette in relazione un cliente registrato ad una o più prenotazioni.
VenditaOnline	Associazione uno-a-molti che mette in relazione un cliente registrato ad una o più vendite online.
Mostrato	Associazione uno-a-molti che mette in relazione una proiezione in una o più sale.
Affiliazione	Associazione uno-a-molti che mette in relazione una proiezione a più biglietti.
Feedback	Associazione molti-a-molti che mette in relazione più clienti a più film.
Precedente	Associazione uno-a-molti che mette in relazione più anteprime ad un film.
Trasmissione	Associazione uno-a-molti che mette in relazione un film a più proiezioni.
Partecipazione	Associazione molti-a-molti che mette in relazione più anteprime a più proiezioni.
Antecedente	Associazione molti-a-molti che mette in relazione più spot a più proiezioni.
Rilascio	Associazione uno-a-uno che mette in relazione una vendita ad una proiezione.

2 Ristrutturazione del modello concettuale

Prima di poter passare allo schema logico è necessario ristrutturare il diagramma delle classi per semplificare la traduzione in schema logico, ottimizzare il progetto, eliminare le generalizzazioni, eliminare gli attributi multivalore, eliminare gli attributi strutturati, accorpate o partizionare le entità figlie e scegliere gli identificatori delle nostre entità ove necessario.

2.1 Analisi delle Ridondanze

Un'operazione della nostra base dati è la gestione delle programmazioni dei film nel cinema multisala. Ciò comporta il sapere una anteprima di un film a quale film si riferisce, quindi il nome del suo trailer, l'anno di creazione del film e il suo regista, motivo per la quale li inseriamo come attributi nella classe *Anteprima*.

2.2 Analisi delle Generalizzazioni

Rimuoviamo la specializzazione totale disgiunta di *Vendita* incorporando le entità *figlie* nell'entità *padre*, inserendo un attributo "*TipoVendita*".

2.3 Eliminazione degli attributi multivalore

Nell'entità *Film* abbiamo un attributo multivalore che ci permette di rappresentare gli attori che recitano in un film. Decido così di creare un'entità a parte chiamata *Attore* utile a mappare il singolo attore, e mediante una relazione, verrà legata ai film nella quale reciterà.

2.4 Eliminazione degli attributi strutturato

In nessuna delle entità ho utilizzato un attributo strutturato.

2.5 Analisi di Entità e Associazioni

Sarà utile rendere *Sala* entità forte e non debole in modo che possiamo far sì che *Posto* sia una sua entità debole, ma *Sala* non lo sarà per *Cinema*. Così facendo non avremo più il problema di identificare *Posto* tramite *Cinema* passando per *Sala*, ma ci basta sapere l'identificativo di una sala.

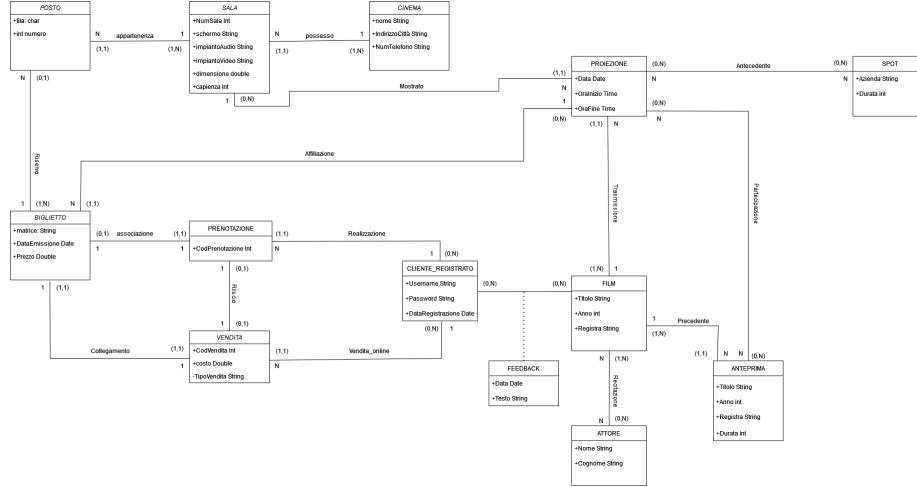
2.6 Identificazione chiavi primarie

Rendendo *Sala* una entità forte, inseriremo una chiave primaria "*NumSala*" per poter identificare la singola sala all'interno di un cinema multisala.

Utile sarà anche inserire una PK di nome *CodPrenotazione* all'interno di *Prenotazione* per poter far riferimento facilmente all'entità nel momento in cui dovremo tener traccia di quanti clienti registrati hanno prenotato un biglietto online.

Inserirò anche in *vendita* una PK di nome *CodVendita* per poterci accedere facilmente e calcolare eventualmente l'incasso totale di un film.

2.7 Schema Ristrutturato



3 Traduzione al Modello Logico

Terminata la fase di ristrutturazione del DB possiamo passare alla traduzione del nostro schema concettuale in uno schema logico. Effettueremo il Mapping delle entità e delle associazioni.

- **CINEMA**(Nome, IndirizzoCittà, NumeroTelefono);
- **SALA**(NumSala, NomeCinema ↑, IndirizzoCittàCinema ↑, Schermo, ImpiantoAudio, ImpiantoVideo, Dimensione, Capienza*);
- **POSTO**(Fila, Numero NumSala ↑);
- **BIGLIETTO**(Matrice, DataEmissione, Prezzo);
- **RISERVO**(Fila, Numero, Matrice);
- **PRENOTAZIONE**(CodPrenotazione, MatriceBiglietto ↑);
- **VENDITA**(CodVendita, MatriceBiglietto ↑, Costo, TipoVendita);
- **RILASCIO**(CodPrenotazione, CodVendita);
- **PROIEZIONE**(Data, OraInizio, OraFine, TitoloFilm ↑, AnnoFilm ↑);
- **AFFILIAZIONE**(MatriceBiglietto, DataProiezione, OraInizioProiezione, OraFineProiezione);
- **MOSTRATO**(DataProiezione, OraInizioProiezione, OraFineProiezione, NumSala);

- **REALIZZAZIONE**(CodPrenotazione, UsernameUtenteRegistrato);
- **VENDITA_ONLINE**(CodVendita, Username);
- **CLIENTE_REGISTRATO**(Username, Password, DataRegistrazione);
- **FILM**(Titolo, Data, Regista);
- **FEEDBACK**(Username, Titolo, Data, Testo);
- **ATTORE**(Nome, Cognome);
- **RECITAZIONE**(Nome, Cognome, Titolo, Anno);
- **ANTEPRIMA**(Titolo, Anno, TitoloFilm↑, AnnoFilm↑, Regista, Durata);
- **PARTECIPAZIONE**(Titolo, Anno, Data, OraInizio, OraFine);
- **SPOT**(Azienda, Durata);
- **ANTECEDENTE**(Azienda, Data, OraInizio, OraFine);