

CinePlace+

Bando Matteo, Da Re Gabriele

29 maggio 2022



1 Abstract

"CinePlace+" è una catena di cinema che conta più di 300 cinema dislocati in tutta Italia. Di ogni cinema si vogliono sapere, per scopi statistici e di gestione del personale, gli incassi totali. Si vuole produrre una base di dati per gestire le sale di ogni cinema, i film proiettati in sala e le prenotazioni degli utenti registrati al portale CinePlace+.

I biglietti vengono acquistati per un certo film ad una certa ora, attraverso un metodo di pagamento che sarà poi associato al biglietto in modo da storicizzare le varie entrate e non avere "pagamenti fantasma". Il metodo viene specificato ogni volta all'acquisto del biglietto.

Non esistono cinema nella stessa città con lo stesso nome e ognuno presenta almeno una sala, la quale può essere di diversi tipi: IMAX, 3D oppure normale. Ogni sala ha anche un predeterminato numero di posti a sedere, invariabili nel tempo.

I film vengono trasmessi per un prestabilito periodo e il medesimo film può essere riprodotto in più sale dello stesso cinema in diversi orari prefissati. Ogni sala in un giorno può avere in programma più film riprodotti ovviamente in orari diversi. Sempre per scopi statistici si vuole memorizzare la media dei voti dei film e delle sale calcolabile attraverso le recensioni degli account premium

2 Analisi dei requisiti

2.1 Descrizione testuale

Ogni **cinema** CinePlace+ è caratterizzato da:

- Nome
- Città
- Indirizzo
- Incassi totali

La **sala** è legata al cinema in cui è posta ed è caratterizzata da:

- IdSala
- Posti massimi
- Prezzo

Ogni sala può essere di 3 tipi: **3D, IMAX o normali** per le sale 3D il prezzo del biglietto è 15 euro, per le IMAX 10 euro e per le normali 7 euro.

Il **film** è riprodotto in una o più predeterminate sale ad una certa ora ed è rappresentato da:

- Nome Film
- Anno di uscita
- Durata

Le **sale ed i film** possono essere recensite da utenti premium che hanno acquistato il biglietto, ogni recensione per film e sale sono individuate da:

- Valutazione
- Commento
- Data

I biglietti possono essere acquistati solo ed esclusivamente online, ergo l'acquirente dovrà prima, mediante portale "CinePlus+", crearsi un **account** descritto da:

Una **persona** la quale sarà caratterizzata da:

- Nome
- Cognome
- Sesso
- Data di nascita
- Codice fiscale
- Numero di telefono

La persona sarà a sua volta collegata univocamente a un **account** individuato da:

- Indirizzo email
- Password

Un utente può acquistare un **biglietto** il quale sarà quindi collegato in modo univoco al profilo dell'acquirente ed è caratterizzato da:

- ID del biglietto
- Data di acquisto

I pagamenti vengono autorizzati attraverso un sito esterno.

Una volta effettuato il pagamento attraverso un popup sul sito, fornito da un servizio da parte di terzi, verrà tornato dalla API del servizio, un valore il quale sarà verificato dalla back-end. Se il pagamento sarà accettato allora verrà inserito il biglietto nel DB.

2.2 Glossario termini

Termine	Descrizione	Collegamenti
Cinema	Sede fisica del cinema nella città ad un certo indirizzo	Sala
Incassi totali	Incassi cumulati da un cinema in un anno	Attributo di Cinema
Sala	Sala presente in una certa sede "Cinema", con un certo numero di posti.	Tariffa, Biglietto, Film, Cinema
Imax	Tipologia di sala	Entità figlia di Sala
3D	Tipologia di sala	Entità figlia di Sala
Posti massimi	Numero di posti a sedere in una certa sala	Attributo di Sala
Film	Film riprodotti in una certa sala per un dato periodo ad un certo orario	Biglietto, Recensione Film, Sala
Recensione	Insieme delle recensioni fatte da un account	Sala, Premium
Recensione film	Insieme delle recensioni sui film fatte da un account	Entità figlia di Recensione, Film
Recensione sala	Insieme delle recensioni sulle sale fatte da un account	Entità figlia di Recensione, Sala
Persona	Entità con dati anagrafici dell'intestatario dell'account	Account
Account	Insieme dei dati necessari per la "gestione della persona" all'interno del sito	Biglietto
Premium	Account con abbonamento mensile abilitato alla creazione di recensioni	Recensione
Biglietto	Biglietto acquistato da un account per un certo spettacolo in un certo posto in sala	Sala, Film, Account

2.3 Operazioni

Operazione	Lettura/Scrittura	Frequenza
Inserimento biglietto per un certo film	S	1000 al giorno
Inserimento account, ovvero profilo utente + inserimento dati per l'account	S	100 al giorno
Inserimento film nella programmazione delle sale di un cinema	S	1 al giorno
Inserimento di una sede in una nuova città	S	~1 anno
Lettura dei biglietti acquistati per uno spettacolo (con relativa ora, data e film)	L	1000 al giorno
Visualizzare il punteggio medio delle recensioni dei film	L	1500 al giorno
Visualizzare il punteggio medio delle recensioni delle sale	L	1500 al giorno
Visualizzazione degli incassi totali di un cinema	L	~10 al giorno
Aggiornamento degli incassi totali di un cinema	S	1 al giorno(fine giornata)
Inserimento di una recensione di un cinema	S	100 al giorno
Visualizzazione della valutazione media di sale e film	S	20000 al giorno

3 Progettazione Concettuale

3.1 Lista entità

- Cinema
 - nome: varchar(50)
 - città: varchar(50)
 - incassi_totali: varchar(100)
 - indirizzo: attributo composto
 - via: varchar(50)
 - numero: char(5)
 - cap: char(5)
 - stato: varchar(30)
- Film
 - nome_film: varchar(50) primary key
 - anno_uscita: date primary key
 - durata: time
 - valutazione_media: numeric(2,1)
- Sala
 - numero_sala: char(2) primary key
 - nome_cinema: varchar(50) primary key
 - città_cinema: varchar(50) primary key
 - posti_massimi: varchar(10)
 - prezzo: money
 - valutazione_media: numeric(2,1)

L'entità Film si specializza in una sottocategoria con una generalizzazione parziale:

- 3D
- IMAX
- Recensione (da mettere recensione film e sala)
 - account_email: varchar(50) primary key
 - data: timestamp primary key
 - commento: varchar(500)
 - valutazione: smallint

L'entità Recensione si specializza in due sottocategoria con una generalizzazione totale:

- Recensione Sala
- Recensione Film
- Account (da mettere quello premium)
 - email: varchar(50) primary key
 - password:varchar(50)

L'entità Account si specializza in una sottocategoria con una generalizzazione parziale:

- Premium
- Persona
 - CF: char(16) primary key
 - cognome: varchar(50)
 - nome: varchar(50)
 - sesso: char(2)
 - data_nascita: date
 - numero_telefono: varchar(15)
- Biglietto
 - biglietto_id: bigserial
 - data_acquisto: timestamp

Relazione	Entità coinvolte	Descrizione	Attributi
Associazione	Film(0,N) Biglietto(1,1)	Un film può essere associato a più biglietti; un biglietto è associato ad un solo ed unico film	Nessuno
Critica Film	Film(0,N) Recensione Film(1,1)	Un film può avere tante Recensioni Film; una specifica Recensione Film può essere collegata ad un unico Film	Nessuno
Programmazione	Film(0,N) Sala(0,N)	Un film può essere trasmesso in tante sale in un orario definito; una sala può trasmettere diversi film in diversi orari	Orario
Collegamento	Sala(0,N) Biglietto(1,1)	Una sala è collegata ad un biglietto e un biglietto è collegato ad un' unica sala	Posto
Transazione	Account(0,N) Biglietto(1,1)	Un account può comprare da 0 ad un numero arbitrario di biglietti; un biglietto è associato ad un unico account	Nessuno
Appartenenza	Account(1,1) Persona(0,1)	Un account appartiene ad una sola persona; Una persona può avere al più un account, ma può decidere inoltre di cancellarlo	Nessuno
Stilata	Recensione(1,1) Premium(0,N)	Una recensione può essere stilata da uno specifico account; Un account premium può non avere stilato alcuna recensione	Nessuno
Critica Sala	Sala(0,N) Recensione Sala(1,1)	Una sala può non avere alcuna recensione e una specifica recensione sala può essere collegata al più ad una sala	Nessuno
Appartenenza	Cinema(0,N) Sala(1,1)	Una cinema può non avere alcuna sala o diverse sale e una specifica appartiene ad un unico Cinema	Nessuno

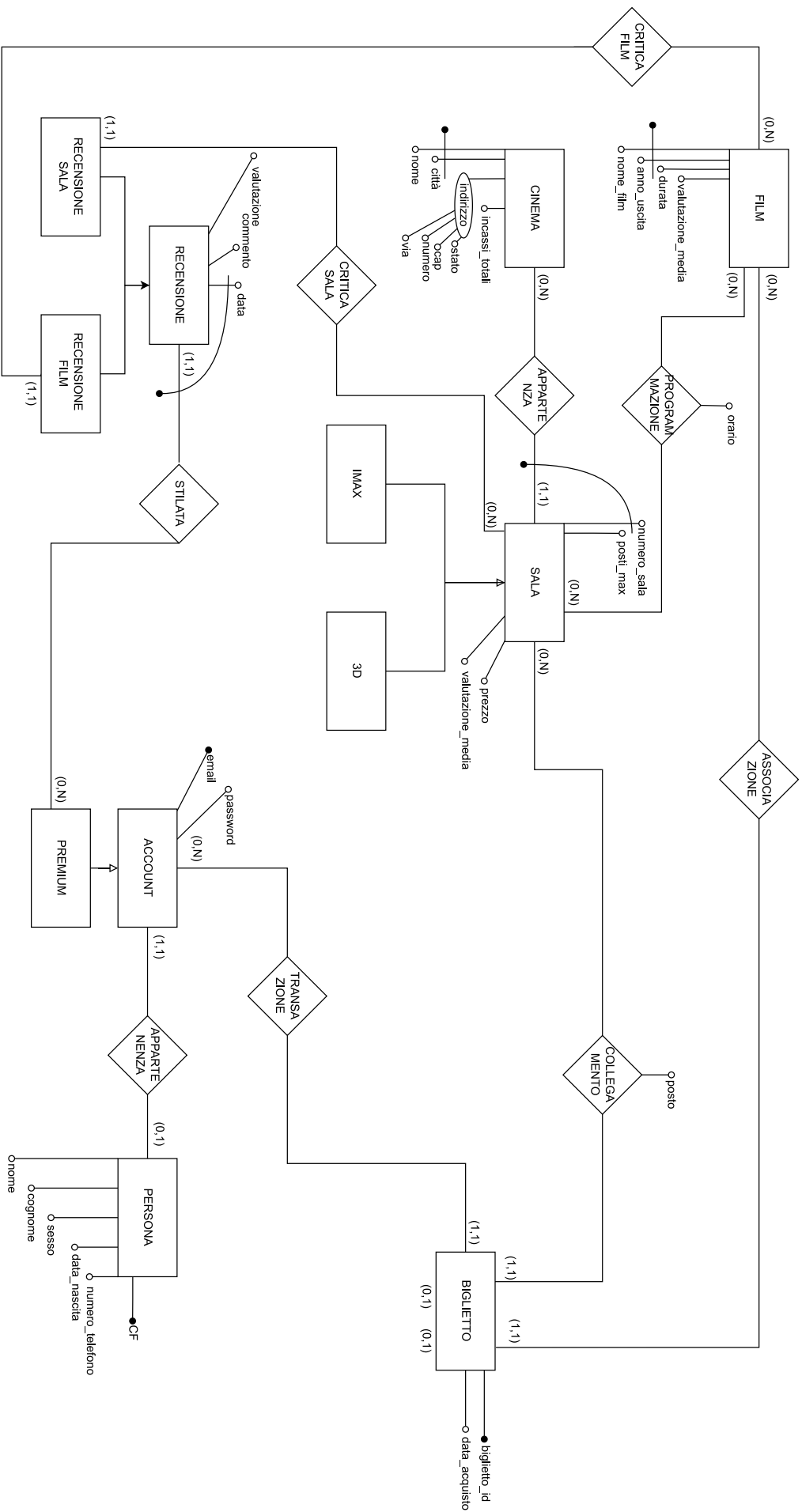
3.2 Tabella delle relazioni

Vincoli non rappresentabili tramite schema E-R:

- La valutazione della sala va da 0 a 9, dove 0 è il minimo e 9 il massimo
- La valutazione del film va da 0 a 9, dove 0 è il minimo e 9 il massimo
- L'attributo "sesso" nella tabella Persona può essere solo di 3 tipi:
 - 1) M - Maschio
 - 2) F - Femmina
 - 3) ND - Not Defined

Vincoli di derivazione:

- L'attributo incassi_totali dell'entità film è uguale alla somma dei biglietti venduti in quel giorno/mese/anno.
- L'attributo valutazione_media dell'entità film e sala è il risultato della media delle recensioni relative rispettivamente al film oppure alla sala.



4 Progettazione Logica

4.1 Ristrutturazione

4.1.1 Analisi delle ridondanze

L'attributo "incassi totali" relativo all'entità "Cinema" è ridondante, in quanto è possibile calcolarlo sommando i prezzi dei biglietti acquistati relativi alle sale di un particolare cinema. Inoltre è necessario solo per altre 2 operazioni:

1. aggiornamento degli incassi totali di un cinema, 2. visualizzazione degli incassi totali di un cinema.

Operazione 1: Eseguita una sola volta per cinema a fine giornata.

Operazione 2: Eseguita circa una decina di volte al giorno quando richiesto dalla front-end.

L'attributo "valutazione_media" delle entità "Film" e "Sala" sono ridondanti in quanto possono essere entrambe calcolate facendo la media dei punteggi delle valutazioni relative rispettivamente ai film e alle sale.

Concetto	Tipo	Volume
Film	E	10000
Recensione Film	E	200000
Critica Film	R	200000

Concetto	Tipo	Volume
Sala	E	10000
Recensione Sala	E	200000
Critica Sala	R	200000

- Operazione 1: inserimento di una nuova recensione. (100 al giorno)
- Operazione 2: visualizzazione della valutazione media di un film/sala (20000 al giorno)

I calcoli delle ridondanze sono i medesimi sia per le recensioni film che per le recensioni sala, in quanto hanno lo stesso volume indicativo, dovuto al fatto che i film totali trasmessi dal cinema sono circa 10^3 come anche per le sale, in quanto tra tutti i cinema della catena si conta circa lo stesso numero di sale.

PRESENZA RIDONDANZA:

Operazione 1

Un film ha mediamente 200000 recensioni film/10000 film = 20 recensioni

Concetto	Costrutto	Accessi	Tipo	Numero di occorrenze
Critica Film	Relazione	1	S	x100
Critica Film	Relazione	20	L	x100
Film	Entità	1	S	x100

Operazione 2

Concetto	Costrutto	Accessi	Tipo	Numero di occorrenze
Film	Entità	1	S	x20000

Considerando gli accessi in scrittura con costo doppio rispetto alla lettura:
costo totale = $100 \cdot 2 + 20 \cdot 100 + 100 \cdot 2 + 20000 = 22400$

ASSENZA RIDONDANZA:

Operazione 1

Concetto	Costrutto	Accessi	Tipo	Numero di occorrenze
Critica Film	Relazione	1	S	x100

Operazione 2

Concetto	Costrutto	Accessi	Tipo	Numero di occorrenze
Film	Entità	20	S	x20000

Costo totale = $100 \cdot 2 + 20 \cdot 20000 = 400200$

In conclusione l'analisi della ridondanza dimostra che conviene mantenere l'attributo "valutazione_media" che verrà aggiornato con l'aggiunta di ogni recensione.

4.1.2 Eliminazione delle generalizzazioni

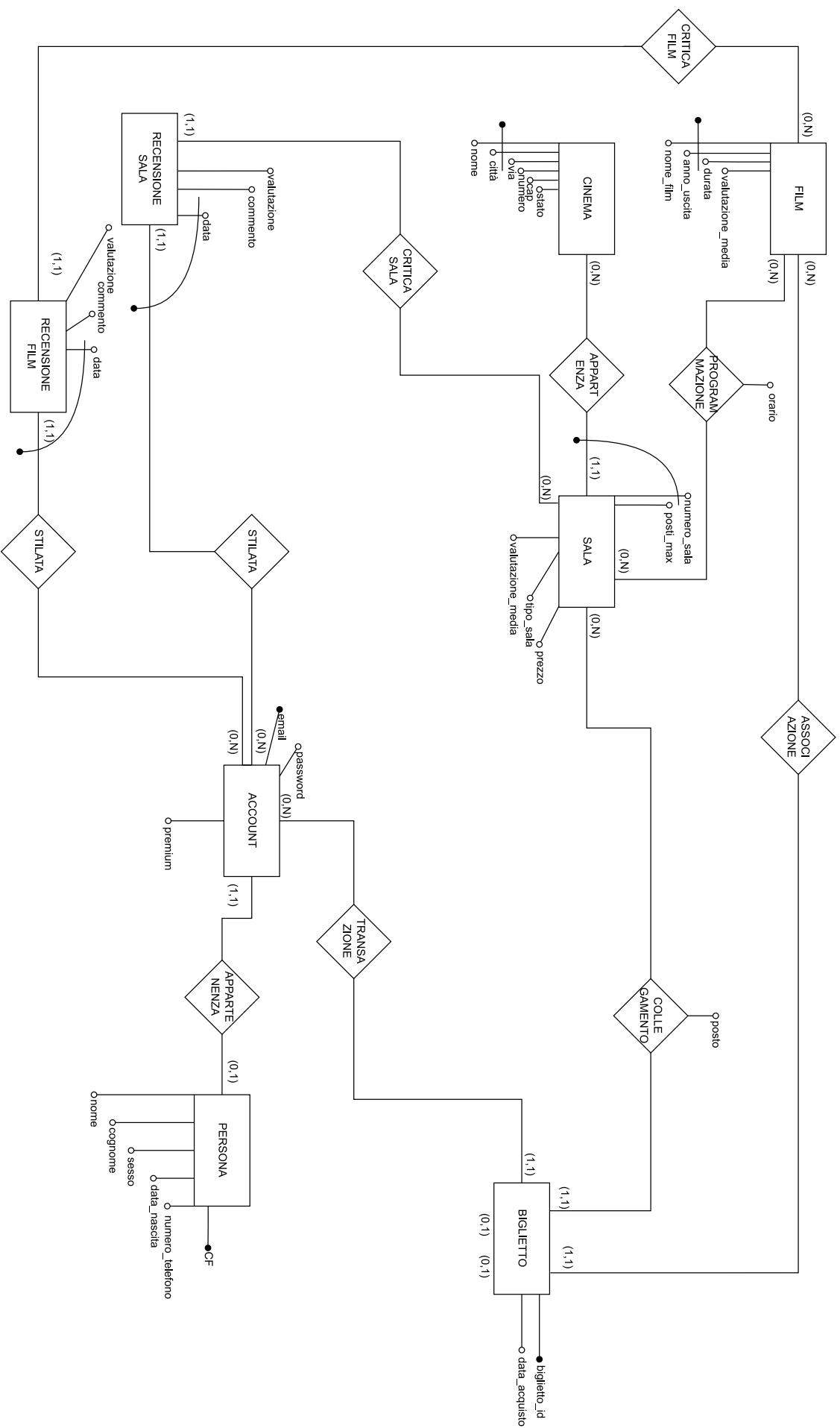
⇐ indica la generalizzazione parziale

← indica la generalizzazione completa

Relazione	Entità coinvolte
Account ⇐ Premium	L'entità Premium viene accorporata in Account, in quanto non presenta nessun attributo particolare che serve alla relazione "scritta"
Recensione ← Recensione Film, Recensione Sala	Il padre Recensione viene accorporato in Recensione Sala e in Recensione Film
Sala ⇐ Imax, 3D	Le figlie vengono accorpate al padre in quanto non hanno alcuna relazione che necessita la generalizzazione

4.1.3 Scelta degli identificatori primari

Schema E-R ristrutturato



4.2 Creazione delle tabelle (A→B indica che B è chiave esterna di A)

film(nome_film, anno_uscita, durata, valutazione_media)

cinema(nome, citta, via, numero, cap, stato)

sala(numero_sala, nome_cinema→cinema.nome, citta_cinema→cinema.citta, posti_massimi, prezzo, tipo_sala, valutazione_media)

recensione_sala(data, email→account.email, commento, valutazione, nome_cinema→sala.nome_cinema, citta_cinema→sala.citta_cinema, numero_sala→sala.numero_sala)

recensione_film(data, email→account.email, commento, valutazione, nome_film→film.nome_film, anno_uscita→film.anno_uscita)

account(email, password, cf→persona.cf, premium)

persona(CF, nome, cognome, sesso, data_nascita, numero_telefono)

biglietto(biglietto_id, data_acquisto, nome_film→film.nome_film, anno_uscita→film.anno_uscita, nome_cinema→sala.nome_cinema, citta_cinema→sala.citta_cinema, numero_sala→sala.numero_sala, email→account.email, posto)

programmazione(nome_film→film.nome_film, anno_uscita→film.anno_uscita, nome_cinema→sala.nome_cinema, citta_cinema→sala.citta_cinema, numero_sala→sala.numero_sala, orario)

Come specificato sopra, verrà mantenuto l'attributo valutazione_media sia per l'entità film che per l'entità sala. Anche se non detto in precedenza, l'implementazione consisterebbe in una trigger function la quale ad ogni nuova recensione inserita nell'entità film (risp. sala) calcola il nuovo valore della valutazione media considerando la nuova recensione, successivamente fa l'update nella tabella film (risp. sala).

5 Query e indici

5.1 Query

1 - Mostra il numero di biglietti comprati da una persona mostrandone NOME COGNOME e CF e indica se può fare o meno recensioni

```
1 SELECT p.nome AS nome, p.cognome, p.cf, a.premium AS puo_recensire, count(*) AS
   biglietti_acquistati
2 FROM persona AS p JOIN account AS a ON p.cf = a.cf JOIN biglietto AS b ON a.email = b.
   email
3 GROUP BY p.nome, p.cognome, p.cf, a.premium
```

	nome	cognome	cf	puo_recensire	biglietti_acquistati
1	Lisa	Babbo	ZMNPTZ82H21B629G	si	10
2	Michelangelo	Brugnaro	RGRLRT56L24M298J	si	10
3	Nicola	Zampa	PRTRS78H13L3590	no	10
4	Priscilla	Cattaneo	FODMA09D54A347EX	si	11
5	Federico	Zabarella	CGNVL750T15G211Y	si	17
6	Martina	Boezio	BRCGNE08M11M180X	si	9
7	Carolina	Broggini	LPPGLR37D07B883S	no	13
8	Delfino	Zamorani	RFFGTN98C65M260G	no	8
9	Ornella	Sforza	SLVGBL37D47H270I	si	13
10	Ludovica	Argurio	BRGGLI78B51M130S	no	8

2 - Mostra tutti i cinema con incassi totali maggiori di 100\$

```
1 SELECT c.nome, c.citta, SUM(prezzo) AS incassi_totali
2 FROM cinema AS c JOIN sala AS s ON c.nome = s.nome_cinema AND c.citta = s.citta_cinema
3 JOIN biglietto AS b ON s.numero_sala = b.numero_sala AND s.nome_cinema = b.nome_cinema AND s
   .citta_cinema = b.citta_cinema
4 GROUP BY c.nome, c.citta
5 HAVING SUM(prezzo) > money(100)
6 ORDER BY incassi_totali
```

	nome	citta	incassi_totali
1	Cinema8	Citta8	\$101.00
2	Cinema9	Citta9	\$107.00
3	Cinema1	Citta1	\$114.00
4	Cinema6	Citta6	\$138.00
5	Cinema4	Citta4	\$157.00
6	Cinema7	Citta7	\$165.00

3 - Trovare tutte le persone che si sono sedute sullo stesso posto (anche in sale diverse)

```
1 SELECT p.nome, p.cognome, a.email
2 FROM persona AS p, account AS a
3 WHERE a.email IN (SELECT email
4 FROM biglietto
5 GROUP BY posto, email
6 HAVING count(biglietto_id)>1) AND p.cf = a.cf
7 GROUP BY p.cf, a.email
```

```
1 +-----+-----+-----+
2 | nome      | cognome  | cf      |
3 +-----+-----+-----+
4 | Carolina  | Brogгинi | LPPGLR37D07B883S |
5 | Federico  | Zabarella | CGNVLT50T15G211Y |
6 +-----+-----+-----+
```

4 - Trovare le sole persone che non hanno scritto una recensione

```
1 DROP VIEW IF EXISTS p_a;
2 CREATE VIEW p_a AS
3 SELECT p.nome, p.cognome, p.cf, a.email
4 FROM persona AS p, account AS a
5 WHERE p.cf=a.cf;
6
7 SELECT nome, cognome, cf
8 FROM p_a, biglietto AS b
9 WHERE p_a.email NOT IN (SELECT email
10 FROM recensione_film as f
11 GROUP BY email) AND p_a.email NOT IN (SELECT email
12 FROM recensione_sala as f
13 GROUP BY email)
14 GROUP BY nome, cognome, cf
```

```
1 +-----+-----+-----+
2 | nome      | cognome  | cf      |
3 +-----+-----+-----+
4 | Priscilla  | Cattaneo | FODMA09D54A347EX |
5 | Ludovica   | Argurio  | BRGGLI78B51M130S |
6 | Nicola     | Zampa    | PRTRSN78H13L3590 |
7 | Delfino    | Zamorani | RFFGTN98C65M260G |
8 | Carolina   | Brogгинi | LPPGLR37D07B883S |
9 +-----+-----+-----+
```

5 - Trovare l'utente che ha acquistato più biglietti con il loro relativo numero

```
1 SELECT p.nome, p.cognome, p.email, count(*) as Numero_biglietti_acquistati
2 FROM p_a p join biglietto b on p.email=b.email
3 GROUP BY p.nome, p.cognome, p.email
4 ORDER BY Numero_biglietti_acquistati DESC
5 LIMIT 1
```

```
1 Risultato delle tuple consistente
2 +-----+-----+-----+-----+
3 | nome      | cognome  | email      | numero_biglietti_acquistati |
4 +-----+-----+-----+-----+
5 | Federico  | Zabarella | ipiccinni@outlook.com | 17 |
6 +-----+-----+-----+-----+
```

6 - Trovare tutti i cinema e le relative sale che riproducono il film "Avengers: Endgame"

```
1 SELECT c.nome, c.citta, s.numero_sala
2 FROM cinema c, sala s, programmazione p, film f
3 WHERE s.nome_cinema = c.nome AND s.citta_cinema = c.citta
4 AND s.nome_cinema = p.nome_cinema AND s.citta_cinema = p.citta_cinema AND p.
   numero_sala = s.numero_sala
5 AND f.nome_film = p.nome_film AND f.anno_uscita=p.anno_uscita
6 AND f.nome_film = 'Avengers: Endgame';
```

```
1 +-----+-----+-----+
2 | nome      | citta    | numero_sala |
3 +-----+-----+-----+
4 | Cinema1   | Citta1   | 8            |
5 | Cinema5   | Citta5   | 5            |
6 +-----+-----+-----+
```

5.2 Indici

Le recensioni dei film e delle sale sono dei dati che vengono utilizzati molto in lettura in quanto servono per tenere aggiornati i valori "valutazione_media" nelle tabelle Film e Sala, ergo si decide, ipotizzando uno sviluppo su larga scala delle tabelle delle recensioni di creare i seguenti indici per ottimizzare i tempi:

```
1 create index film_recensioni on recensione_film(nome_film, anno_uscita, valutazione);
```

```
1 create index sala_recensioni on recensione_sala(numero_sala, nome_cinema, citta_cinema,
    valutazione);
```

6 Codice C++

6.1 Descrizione dell'utilizzo del codice

Per creare l'eseguibile dal file .cpp occorre compilarlo con il seguente comando:

```
1 g++ main.cpp -L dependencies\lib -lpq -o main
```

Prima di compilare il file è necessario spostare nella cartella **lib** i file

libpq.dll
libpq.lib

e nella cartella **include** i file:

libpq-fe.h
pg_consig_ext.h
postgres_ext.h

Inoltre sarà necessario [aggiungere alle variabili d'ambiente di Windows](#) i seguenti percorsi:

C:\Program Files\PostgreSQL\pg_version\bin
C:\Program Files\PostgreSQL\pg_version\lib

nel nostro caso pg_version = 14

All'apertura verrà mostrato un menù con diverse scelte in base all'input dell'utente:

- 1 - Crea le tabelle all'interno del DB
- 2 - Popola il DB
- 3 - Mostra il numero di biglietti comprati da una persona mostrandone NOME COGNOME e CF e indica se può fare o meno recensioni
- 4 - Mostra tutti i cinema con incassi totali maggiori di 100\$
- 5 - Trovare tutte le persone che si sono sedute sullo stesso posto (anche in sale diverse)
- 6 - Trovare le sole persone che non hanno scritto una recensione
- 7 - Trovare l'utente che ha acquistato più biglietti con il relativo numero
- 8 - Trovare tutti i cinema e le relative sale che riproducono il film "Avengers: Endgame"
- 9 - Stampa sul file "query3.csv" il risultato della query numero 3

6.2 Documentazione del codice

Funzioni implementate ed utilizzate nel codice:

```
1 PGconn* connect(const char* connInfo);
```

Inizializza la connessione al database attraverso i parametri passati tramite connInfo e verifica lo stato della connessione. Per maggiori informazioni visitare [PQConnectDB](#).

```
1 void abort(PGconn* conn = nullptr, PGresult* res = nullptr);
```

Passati una connessione ad un database ancora attiva e il risultato di una query, libera la memoria dalla query e chiude la connessione. Per ulteriori info [PQfinish](#) [PQclear](#).

```
1 string queryReader(const string& path, const string& nome_file);
```

Usa la libreria standard e string per leggere un file ".sql" con un certo nome e path specificati nei parametri nome_file e path.

```
1 PGresult* execute(PGconn* conn, const char* query);
```

Passati una connessione valida ed una query da eseguire (che produca o meno un output), sfruttando il tipo `enum ExecStatusType` della libreria `libpq`, stampa a video lo stato dell'esecuzione della query. Per più info [PQexec](#). [PQresultStatus](#).

```
1 void printSeparator(int campi, int* maxChar);
```

Separatore per la tabulazione in bash.

```
1 void printQuery(PGresult* res);
```

Funzione per stampare nella bash la query risultante l'interrogazione del database.

```
1 char menu(const char* connInfo);
```

Permette, una volta passate le informazioni per la connessione al DB, di eseguire una query oppure di resettare el tabelle ai valori originari.

```
1 void fprintf(PGresult *res, const string& nome_file, const string& path="");
```

Passati il risultato della query, il nome del file in cui stampare la query e il path in cui salvare il file, salva il risultato in formato csv della query all'interno del file nel path specificato.