

---

# Progetto Basi di Dati

---

Damiano Bertoldo

Alessandro Canel

2019/2020

## Contents

<b>1 Abstract</b>	<b>1</b>
<b>2 Analisi dei requisiti</b>	<b>1</b>
2.1 Lista delle operazioni . . . . .	2
2.2 Glossario dei termini . . . . .	3
2.3 Strutturazione dei requisiti . . . . .	3
<b>3 Progettazione Concettuale</b>	<b>4</b>
3.1 Lista delle entita' . . . . .	4
3.2 Lista delle associazioni . . . . .	6
3.3 Schema ER . . . . .	8
<b>4 Progettazione logica</b>	<b>8</b>
4.1 Analisi ridondanza . . . . .	8
4.1.1 Presenza di ridondanza . . . . .	9
4.1.2 Assenza di ridondanza . . . . .	9
4.2 Ristrutturazione schema ER . . . . .	10
4.2.1 Eliminazione generalizzazioni . . . . .	10
4.2.2 Introduzione entita' indirizzo . . . . .	10
4.2.3 Partizionamento/Accorpamento di entita' e relazioni . . . . .	10
4.2.4 Scelta degli identificatori primari . . . . .	11
4.2.5 Schema ER ristrutturato . . . . .	12
4.3 Traduzione nel modello relazionale . . . . .	12
<b>5 Query e Indici</b>	<b>13</b>
5.1 Query . . . . .	13
5.2 Indici . . . . .	15
5.2.1 Studio caso d'uso . . . . .	15
5.2.2 Riconoscimento tabelle da indicizzare . . . . .	15
5.2.3 Implementazione dell'indice . . . . .	15

# 1 Abstract

La società D4C Cinemas è una catena di cinema, nata in regno unito nel 1988 da una partnership tra Universal Studios e Paramount Pictures che nel corso degli anni si è estesa sia nel campo nazionale, che nel campo internazionale.

Si vuole gestire la base di dati del cinema, comprendente la gestione di più cinema contemporaneamente. Ogni cinema è un multisala contenente dalle 5 alle 10 sale ed ogni sala ha dai 50 ai 250 posti. D4C Cinemas rilascia giornalmente la programmazione degli spettacoli in modo tale da dare ai clienti varia scelta per l'orario che desiderano. Una persona per acquistare un biglietto deve recarsi alla biglietteria e al momento dell'acquisto comunicare le credenziali, dato che il biglietto è nominativo a differenza di altri cinema. Ogni biglietto ha un prezzo fisso di 10 euro. Altra caratteristica di questa catena è che al momento dell'uscita dalla sala lo spettatore può lasciare una recensione a caldo del film appena visto, così da lasciare ai futuri spettatori la propria opinione. Altra cosa che viene effettuata in questa catena è l'aggiunta e il tracciamento dei cast presenti, in modo tale da dare al cliente la possibilità di avere la lista di film in cui recita il proprio attore/regista/sceneggiatore/produttore preferito.

## 2 Analisi dei requisiti

Il progetto vuole rappresentare una base di dati che permetta di gestire la programmazione dei film della catena D4C cinema.

Essendo D4C cinemas una catena, è necessario identificare ciascun **Cinema**:

- Nome cinema
- Indirizzo, così composto: via, città, numero civico, codice postale, nazione

Per poter identificare una **Sala** all'interno del cinema abbiamo bisogno delle seguenti informazioni:

- Numero sala (univoco all'interno del cinema)
- Cinema di appartenenza
- Numero posti
- Grandezza schermo (espresso in pollici)

Per ogni Sala vengono proiettati vari film durante la giornata. Tali orari sono definiti nell'entità **Spettacolo**:

- Film
- Sala di proiezione
- Cinema
- Data ora film
- 3D
- Posti venduti

I clienti del cinema durante la scelta del film devono avere la possibilità di consultare le informazioni dei vari **Film**, dati necessari anche per l'organizzazione degli spettacoli:

- Titolo
- Durata
- Trama
- Genere

- Anno

Ciascun film è prodotto da un cast **Cast**, di cui si conosce:

- Nome

Per sapere da chi è composto il cast del film, è necessario avere nome e ruolo delle persone che ne hanno fatto parte, dette persone fanno parte dell'entità **Lavoratore**, e viene inteso come persone che hanno lavorato con mansioni in ambito cinematografico:

- Persona
- Cast
- Salario

Lavoratore a sua volta ha come sottocategoria le seguenti mansioni, dato che lavoratore viene inteso come persona che ha partecipato in mansioni per la realizzazione di un film:

- **Attore**: persona che ha recitato nel film
- **Sceneggiatore**: persona che ha strutturato la sceneggiatura del film
- **Produttore**: persona che ha finanziato il film
- **Regista**: persona che ha diretto il film

Di conseguenza è necessario definire **Persona**, in quanto ogni **Lavoratore** è per l'appunto, una persona e quindi tale entità dovrà comprendere:

- Nome
- Cognome
- Data di nascita
- Sesso

Ogni persona in quanto tale, potrà acquistare un biglietto per un qualsivoglia film. Il **Biglietto** conterrà:

- Persona di appartenenza
- Spettacolo
- Numero posto

Qualsiasi persona potrà inoltre rilasciare una **Recensione** inerente al film appena visto. L'entità recensione conterrà:

- Persona
- Film
- Commento
- Voto

## 2.1 Lista delle operazioni

**Operazione 1:** Aggiunta di uno spettacolo per la proiezione di un film in una sala (40 volte al giorno)

**Operazione 2:** Emissione biglietti (2500 volte al giorno)

**Operazione 3:** Aggiunta nuovi film (5 volte al mese)

**Operazione 4:** Rilascio di una recensione (2200 volte al giorno)

**Operazione 5:** Aggiunta di un nuovo cast (4 volte al mese)

**Operazione 6:** Conteggio totale posti venduti per ogni spettacolo (1 volta a settimana)

## 2.2 Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Spettacolo	Manifestazione artistica o ricreativa presentata a un pubblico		Biglietto, Film, Sala
Sala	Stanza dove viene proiettato il Film		Cinema, Spettacolo
Biglietto	Acquisto del cliente, che attesta la vendita del posto per un dato film		Spettacolo, Persona
Lavoratore	Persona che lavora all'interno di un cast	Attore, Sceneggiatore, Produttore, Regista	Cast, Persona
Cast	Gruppo di lavoratori che ha preso parte alla realizzazione del film		Lavoratore, Film
Persona	Persona cliente del cinema e/o lavoratore		Recensione, Biglietto
Film	Film presenti al cinema		Cast, Recensione, Spettacolo
Cinema	Impresa che offre il servizio cinematografico		Sala
Recensione	Recensioni da clienti paganti		Film, Persona

## 2.3 Strutturazione dei requisiti

<b>Fraasi relative a "Cinema"</b>
I Cinema vengono identificati tramite: Nome cinema, Indirizzo (composto da via, città, numero civico, codice postale, nazione)
<b>Fraasi relative a "Sala"</b>
Le Sale vengono identificate tramite: Numero sala (univoco all'interno del cinema), Cinema di appartenenza, Numero posti, Grandezza schermo (espresso in pollici)
<b>Fraasi relative a "Spettacolo"</b>
Lo spettacolo viene identificato tramite: Film, Sala, Cinema, Data e ora inizio spettacolo, Se il film è 3D, posti venduti
<b>Fraasi relative a "Film"</b>
I Film vengono identificati tramite: Titolo, Durata, Trama, Genere, Anno

<b>Frase relative a "Cast"</b>
Il Cast viene identificato tramite: Nome
<b>Frase relative a "Lavoratore"</b>
I Lavoratori vengono identificati tramite: Persona, Cast, Salario
<b>Frase relative a "Persona"</b>
Le Persone vengono identificate tramite: Nome, Cognome, Data nascita e Sesso
<b>Frase relative a "Recensione"</b>
Le recensioni vengono identificati tramite: Persona, Film, Commento, Voto
<b>Frase relative a "Attore"</b>
Lavoratore che ha lavorato come attore nel film
<b>Frase relative a "Sceneggiatore"</b>
Lavoratore che ha lavorato come sceneggiatore nel film
<b>Frase relative a "Produttore"</b>
Lavoratore che ha lavorato come produttore nel film
<b>Frase relative a "Regista"</b>
Lavoratore che ha lavorato come regista nel film
<b>Frase relative a "Biglietto"</b>
I Biglietti vengono idetificati tramite: Persona, Spettacolo, Numero posto

### 3 Progettazione Concettuale

#### 3.1 Lista delle entita'

Di seguito verranno riportate le classi che verranno contenute nel database, tutti i campi sono **not null** tranne quelli specificati.

- **Cinema:** Rappresenta un cinema.
  - Nome: *string* univoco
  - Indirizzo: attributo composto  
Via: *string*

Citta': *string*  
Provincia: *string*  
CAP: *int*  
Regione: *string*  
Nazione: *string*

- **Sala:** Rappresenta una sala di un cinema.
  - Numero: *int*
  - Cinema: *string*
  - Capienza: *int*
  - Grandezza\_schermo: *int*
- **Spettacolo:** Rappresenta dati per la programmazione degli spettacoli.
  - Film: *int*
  - Sala: *int*
  - Cinema: *string*
  - DataOra\_Inizio: *timestamp*
  - 3D: *bool*
  - Posti\_Venduti: *int*
- **Biglietto:** Rappresenta il biglietto venduto.
  - Persona: *int*
  - Spettacolo: *int*
  - N\_Posto: *int*
- **Film:** Rappresenta un film.
  - Titolo: *string*
  - Anno: *int*
  - Durata: *time*
  - Genere: *string*
  - Trama: *string*
- **Cast:** Rappresenta il nome del cast.
  - Nome: *string*
- **Lavoratore:** Rappresenta una persona partecipante nel cast.
  - Persona: *int*
  - Cast: *int*
  - Salario: *int*
- **Attore:** Specializzazione di Lavoratore
  - Nessun Attributo
- **Produttore:** Specializzazione di Lavoratore
  - Nessun Attributo
- **Sceneggiatore:** Specializzazione di Lavoratore

- Nessun Attributo
- **Regista:** Specializzazione di Lavoratore
  - Nessun Attributo
- **Persona:** Rappresenta una persona che acquista i biglietti e/o fa parte della produzione del film
  - Nome: *string*
  - Cognome: *string*
  - Data\_di\_nascita: *datetime*
  - Sesso: *string*
- **Recensione:** Rappresenta la recensione di un film da parte di una persona.
  - Persona: *int*
  - Film: *int*
  - Commento: *string*, possibilita' di null se la persona non rilascia un commento
  - Voto: *int*

### 3.2 Lista delle associazioni

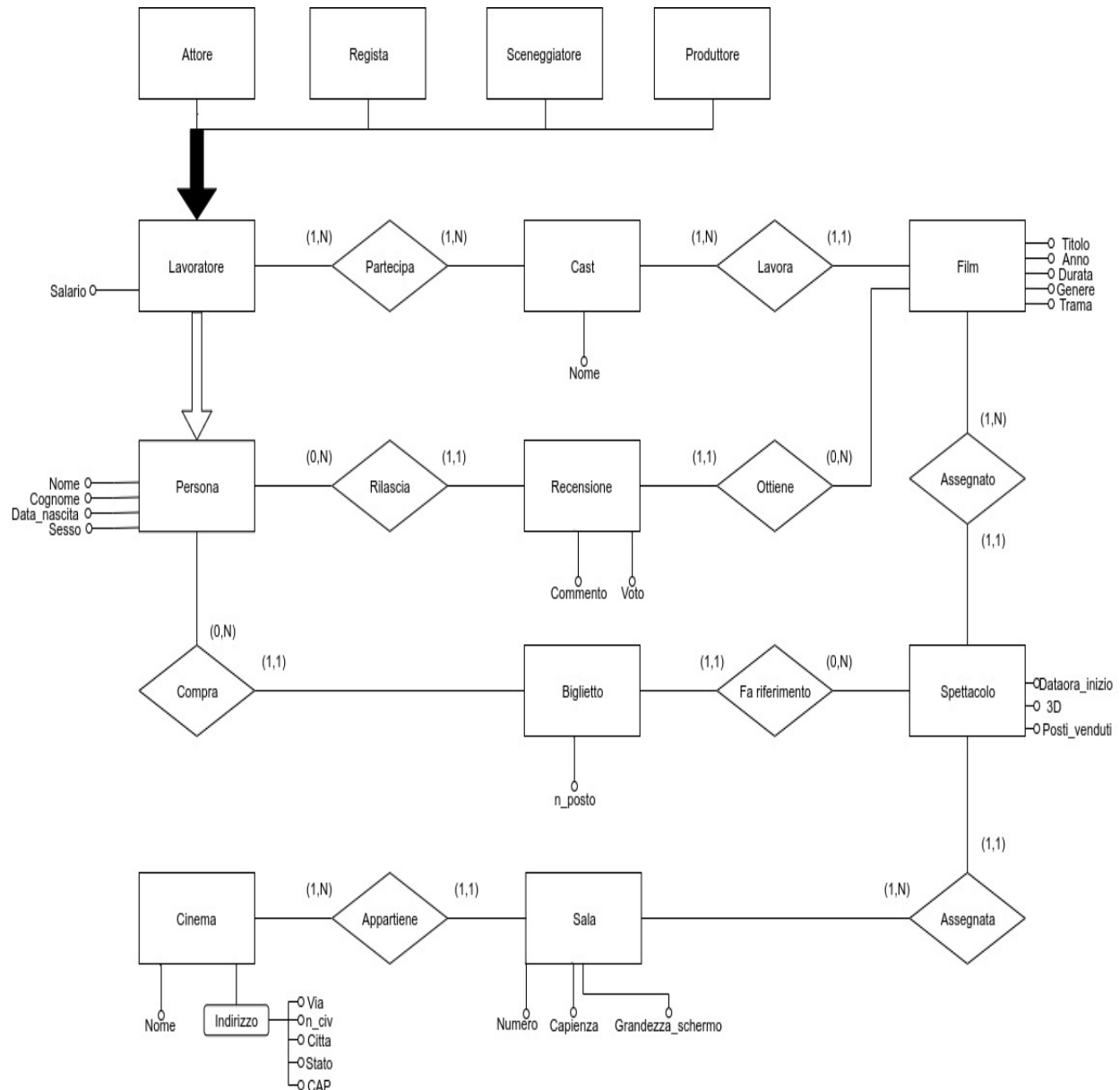
Tutte le entità sono legate nel seguente modo:

- **Cinema-Sala:** Appartiene
  - Ogni cinema può avere diverse sale
  - Ogni sala può appartenere solo ad un cinema
  - Associazione di tipo (1,N) uno a molti
- **Sala-Spettacolo:** Assegnata
  - Ogni sala può aver assegnati diversi spettacoli
  - Ogni spettacolo e' proiettato in una sala
  - Associazione di tipo (1,N) uno a molti
- **Spettacolo-Film:** Assegnato
  - Ogni Spettacolo riproduce un solo Film
  - Un film può essere utilizzato per diversi Spettacoli
  - Associazione di tipo (1,N) uno a molti
- **Film-Cast:** Lavora
  - Ogni Cast può Lavorare a molti Film
  - Un film può avere solo un cast
  - Associazione di tipo (1,N) uno a molti
- **Cast-Lavoratore:** Partecipa
  - Ad un Cast partecipano diversi lavoratori
  - Un lavoratore partecipa a diversi cast
  - Associazione di tipo (N,M) molti a molti

- **Persona-Recensione: Rilascia**
  - Ogni Persona può scrivere diverse Recensioni
  - Ogni Recensione e' scritta da una sola persona
  - Associazione di tipo (1,N) uno a molti
- **Recensione-Film: Ottiene**
  - Ogni Recensione riguarda un solo film
  - Qualsiasi Film può ricevere diverse recensioni da parte delle persone
  - Associazione di tipo (1,N) uno a molti
- **Persona-Biglietto: Compra**
  - Ogni persona può comprare biglietti per vari film
  - Ogni biglietto è intestato ad una sola persona
  - Associazione di tipo (1,N) uno a molti
- **Biglietto-Spettacolo: Fà riferimento**
  - Un biglietto fà riferimento ad un solo spettacolo
  - Per ogni spettacolo vengono comprati diversi biglietti
  - Associazione di tipo (1,N) uno a molti



### 3.3 Schema ER



## 4 Progettazione logica

### 4.1 Analisi ridondanza

Una ridondanza si trova nell'entità **Spettacolo**, nella quale l'attributo **posti\_venduti** si può calcolare visitando la relazione **Spettacolo-Biglietto** (chiamata "fa riferimento") e contando le occorrenze di tale relazione.

- **Operazione 2:** Emissione biglietti (2500 volte al giorno)
- **Operazione 6:** Conteggio totale posti venduti per ogni spettacolo (1 volta a settimana)

L'operazione 6 ha come scopo contare l'affluenza settimanale dei clienti.

Concetto	Tipo	Volume
Spettacolo	E	200
Biglietto	E	28000
Fa' Riferimento	R	28000

Di seguito le operazioni di conteggio in scrittura verranno considerate doppie perche' operazioni piu onerose rispetto alle operazioni di lettura.

#### 4.1.1 Presenza di ridondanza

L'attributo **posti\_venduti**, cosi' come indicato nello schema ER, indica il numero totale di posti venduti per uno spettacolo.

Table 1: **Operazione 2 con ridondanza**

Concetto	Costrutto	Accessi	Tipo
Biglietto	E	1	S
Fa' riferimento	R	1	S
Spettacolo	E	1	L
Spettacolo	E	1	S

- Costi  $3 \cdot 2500 + 1 \cdot 2500 = 17500$  accessi

Table 2: **Operazione 6 con ridondanza**

Concetto	Costrutto	Accessi	Tipo
Spettacolo	E	1	L

- Costi  $1 \cdot 200 = 200$  accessi

Il totale degli accessi, essendo l'operazione 2 giornaliera e l'operazione 6 settimanale, si calcola in questo modo:  $17500 \cdot 7 + 200 = 122700$  accessi.

#### 4.1.2 Assenza di ridondanza

In questo caso per calcolare i posti venduti senza l'attributo **posti\_venduti**, bisogna fare riferimento alla relazione **Spettacolo-Biglietto** (relazione di nome "Fa' riferimento") e contarne le rispettive occorrenze.

Table 3: **Operazione 2 senza ridondanza**

Concetto	Costrutto	Accessi	Tipo
Biglietto	E	1	S
Fa' riferimento	R	1	S

- Costi  $2 \cdot 2500 = 10000$  accessi

Table 4: **Operazione 6 senza ridondanza**

Concetto	Costrutto	Accessi	Tipo
Spettacolo	E	1	L
Fa' riferimento	R	140	L

- Costi  $1 \cdot 200 + 140 \cdot 200 = 28200$  accessi

Il totale degli accessi, essendo l'operazione 2 giornaliera e l'operazione 6 settimanale, si calcola in questo modo:  $10000 \cdot 7 + 28200 = 98200$  accessi.

In conclusione la ridondanza del dato non porta vantaggi, di conseguenza si e' deciso di eliminarla.

## 4.2 Ristrutturazione schema ER

### 4.2.1 Eliminazione generalizzazioni

**Persona** La generalizzazione *Persona*, rappresenta una persona che e' un cliente del cinema, ha come figlia l'entita' *Lavoratore*, che in questo caso viene considerato come il lavoratore che prende parte alla realizzazione di un film.

Essendo una generalizzazione parziale, si e' deciso di dividere le due entita' e creare una associazione tra le due. Questo perche' ci sono entita' che riferiscono singolarmente con l'entita' figlia *Lavoratore*.

**Lavoratore** A sua volta la generalizzazione di *Lavoratore* ha quattro entita' figlie: *Attore*, *Sceneggiatore*, *Produttore* e *Regista*.

Essendo in questo caso una generalizzazione completa, e non essendoci associazioni dirette con le entita' figlie, si e' deciso di accorpate le entita' figlie all'interno dell'entita' *Lavoratore*, aggiungendone un nuovo attributo *Ruolo*, il quale ha lo scopo di identificare i vari ruoli svolti dal lavoratore.

### 4.2.2 Introduzione entita' indirizzo

Per rappresentare le informazioni sull'indirizzo riguardanti la posizione di un *Cinema*, è stata introdotta l'entità *Indirizzo*. Questa entità contiene i seguenti attributi: cap, via, numero civico, nazione e città. *Cinema* sara' l'unica entita' collegata ad *Indirizzo*.

### 4.2.3 Partizionamento/Accorpamento di entita' e relazioni

Nel caso di **Lavoratore**, l'accorpamento della sua generalizzazione *Attore/Sceneggiatore/Produttore/Regista* crea ambiguita' in quanto un lavoratore puo' partecipare a diversi cast con diversi ruoli e salari. Oltre a questo, essendo la relazione *Cast-Lavoratore* una relazione n,m molti a molti, la sua traduzione in modello relazionale si compone in questo modo:

- **Lavoratore:** (Persona, Ruolo, Salario)
- **Cast-Lavoratore:** (Persona, Ruolo, Cast)
- **Cast:** (Id, Nome)

Di conseguenza si e' deciso di accorpate *Cast-Lavoratore* in *Lavoratore* aggiungendo l'attributo **Cast** di tipo **int** e rendendo la relazione *Cast-Lavoratore* di tipo 1,N.

#### 4.2.4 Scelta degli identificatori primari

**Cinema** Come scelta dell'identificatore primario nell'entità, viene scelto l'attributo *Nome*, perché il nome identifica un singolo cinema.

**Sala** Per identificare una sala, viene scelto come identificatore il numero della sala, *numero*, e il cinema di appartenenza, *Cinema*. Questo perché all'interno del cinema il numero della sala è univoco.

**Spettacolo** Per identificare uno spettacolo bisognerebbe fare riferimento all'attributo *dataora\_inizio* insieme alla *sala* e al *film*. Quindi per semplicità viene aggiunto un nuovo attributo identificatore *Id*, cioè semplifica e rende meno ambigua l'identificazione di una tupla all'interno di spettacolo.

**Biglietto** Per identificare il singolo biglietto si utilizza *Persona* e *Spettacolo*, dato che la coppia ne identifica una tupla.

**Film** Per identificare un singolo film, bisognerebbe comprendere *Titolo*, *Anno*, *Genere* e *Durata*, dato che potrebbe verificarsi l'uscita di un film omonimo nello stesso anno con lo stesso genere. Quindi per semplificare e rendere ancora più forte l'identificazione, viene introdotto l'attributo identificatore *Id*.

**Recensione** Per identificare la singola recensione si utilizza *Persona* e *Film*, dato che la coppia ne identifica una tupla.

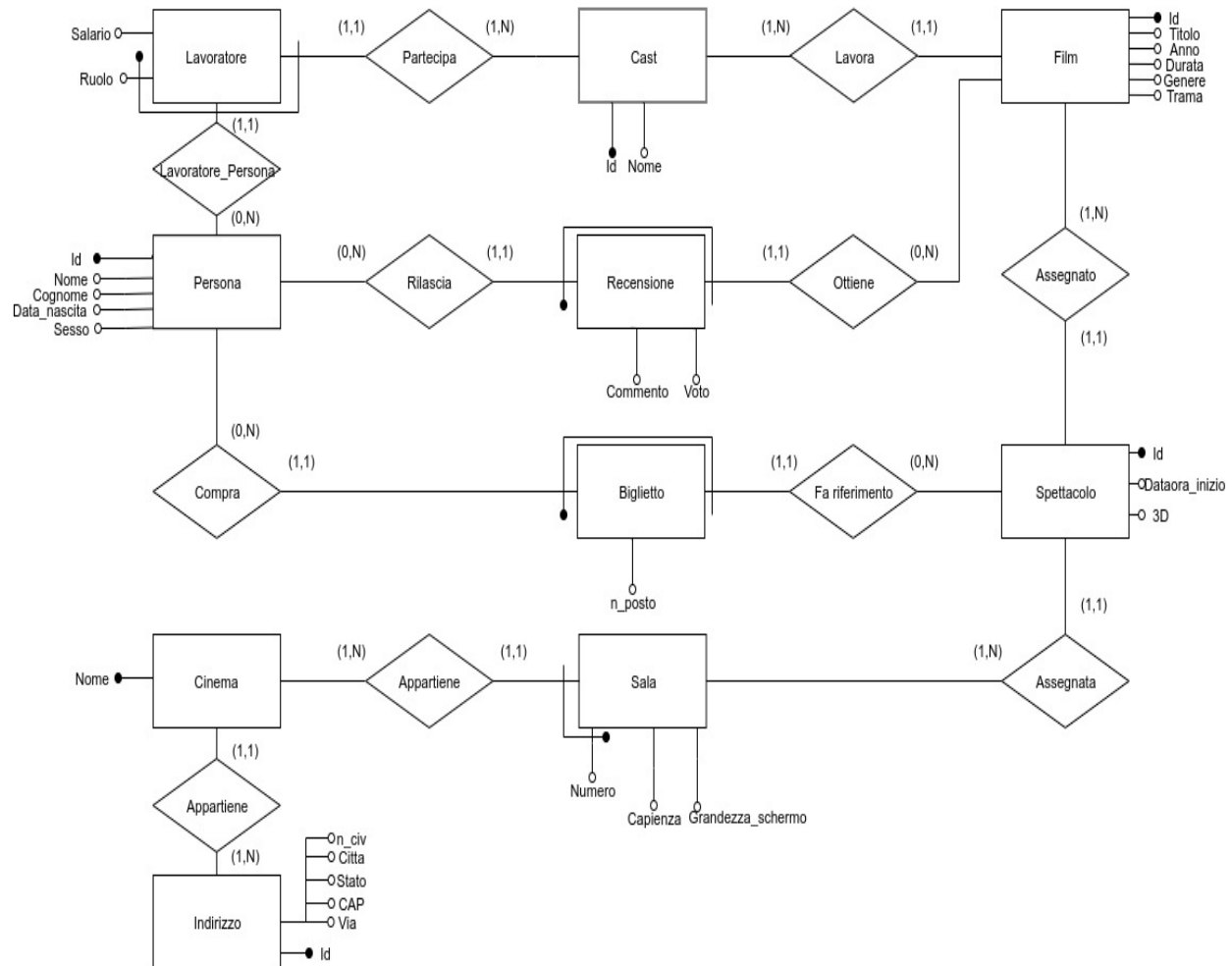
**Persona** Per identificare una persona, *Nome*, *Cognome* e *Data\_nascita* possono risultare inadeguati, quindi viene introdotto l'attributo identificatore *Id*.

**Lavoratore** Per identificare un lavoratore, viene utilizzato il suo *Ruolo*, il riferimento a persona *Persona* e la sua presenza ad un *Cast*, questo perché un lavoratore può assumere più ruoli all'interno di un cast.

**Cast** Per identificare un cast si ricorre all'utilizzo di un identificatore *Id*, questo perché potrebbero esserci cast omonimi.

**Indirizzo** Per identificare un indirizzo si ricorre all'utilizzo di un identificatore *Id*, questo per minimizzare l'utilizzo di attributi per il riconoscimento della singola tupla.

#### 4.2.5 Schema ER ristrutturato



#### 4.3 Traduzione nel modello relazionale

**Cinema**(Nome, Id, Indirizzo)

Vincolo di integrità' referenziale tra "Id\_Indirizzo" ed "Id" in Indirizzo.

**Sala**(Numero, Nome\_Cinema, Capienza, Grandezza\_schermo)

Vincolo di integrità' referenziale tra "Nome\_Cinema" e "Nome" in Cinema.

**Spettacolo**(Id, Dataora\_inizio, 3D, Id\_Film, Num\_Sala, Nome\_Cinema)

Vincolo di integrità' referenziale tra "Id\_Film" ed "Id" in Film, "Num\_Sala" e "Numero" in Sala e tra "Nome\_Cinema" e "Cinema" in Sala.

**Biglietto**(Id\_Spettacolo, Id\_Persona, n\_posto)

Vincolo di integrità' referenziale tra "Id\_Spettacolo" ed "Id" in Spettacolo e tra "Id\_Persona" e "Id" in Persona.

**Film**(Id, Titolo, Anno, Durata, Genere, Trama, Id\_Cast)

Vincolo di integrità' referenziale tra "Id\_Cast" e "Id" in Cast.

**Recensione**(Id\_Film, Id\_Persona, Commento, Voto)

Vincolo di integrità' referenziale tra "Id\_Film" e "Id" in Film e tra "Id\_Persona" e "Id" in Persona.

**Persona**(Id, Nome, Cognome, Data\_nascita)

**Lavoratore**(Id\_Persona, Id\_Cast, Ruolo, Salario)

Vincolo di integrità referenziale tra "Id\_Persona" e "Id" in Persona e tra "Id\_Cast" e "Id" in Cast.

**Cast**(Id, Nome).

**Indirizzo**(Id, Via, Citta, n\_civ, Stato, Cap)

## 5 Query e Indici

### 5.1 Query

**Query1** Numero di recensioni di persone che non hanno visto il film, quindi di conseguenza non valide.

```
1  SELECT p.Id, p.Nome, p.Cognome, COUNT(r.Id_Persona) AS 'N. Recensioni non valide'
2  FROM Recensione r JOIN Persona p ON (r.Id_Persona=p.Id) LEFT JOIN (SELECT DISTINCT f.Id
   AS Id_Film, b.Id_Persona AS Id_Persona, f.Titolo as Titolo
3  FROM Biglietto b, Spettacolo s, Film f
4  WHERE s.Id=b.Id_Spettacolo AND f.Id=s.Id_Film
5  ) sq ON (r.Id_Persona= sq.Id_persona AND r.Id_Film=sq.Id_Film)
6  WHERE sq.Id_Persona IS NULL
7  GROUP BY p.Id
8  --Risultato Query
9  +-----+-----+-----+-----+
10 | Id | Nome      | Cognome  | N. Recensioni non valide |
11 +-----+-----+-----+-----+
12 | 1 | Mario    | Rossi    | 4 |
13 | 2 | Matteo   | Verdi    | 6 |
14 | 3 | Stefano  | Minto    | 6 |
15 | .. | ....    | ...      | ... |
16 +-----+-----+-----+-----+
```

**Query2** Voto medio per film in base alle recensioni valide.

```
1  SELECT sq.Titolo, AVG(r.Voto) AS 'Voto Medio'
2  FROM Recensione r JOIN (SELECT DISTINCT f.Id AS Id_Film, b.Id_Persona AS Id_Persona, f.
   Titolo as Titolo
3  FROM Biglietto b, Spettacolo s, Film f
4  WHERE s.Id=b.Id_Spettacolo AND f.Id=s.Id_Film
5  ) sq ON (r.Id_Persona= sq.Id_persona AND r.Id_Film=sq.Id_Film)
6  GROUP BY sq.Titolo
7  --Risultato Query
8  +-----+-----+-----+
9  | Titolo          | Voto Medio |
10 +-----+-----+-----+
11 | Cinepanettone   | 3.5000 |
12 | Jumanji         | 1.6667 |
13 | Mediterraneo    | 4.0000 |
14 | ...             | ...      |
15 +-----+-----+-----+
```

### Query3 Affluenza di persone per cinema.

```
1 SELECT s.Nome_Cinema AS Cinema, COUNT(b.Id_Persona) AS Affluenza
2 FROM Biglietto b JOIN Spettacolo s ON (b.Id_Spettacolo=s.Id)
3 GROUP BY s.Nome_Cinema
4 ORDER BY s.Nome_Cinema
5 --Risultato Query
6 +-----+-----+
7 | Cinema      | Affluenza |
8 +-----+-----+
9 | D4C Marcon   |         10 |
10 | D4C Mestre   |          4 |
11 | D4C Milano   |          4 |
12 | ...          | ...       |
13 +-----+-----+
```

### Query4 Lista attore/sceneggiatore/produttore/regista piu' pagato.

```
1 SELECT sq.Id_Persona AS ID, p.Nome, p.Cognome, p.Sesso, sq.Ruolo, MAX(sq.Salario) AS
   Salario
2 FROM (SELECT Id_Persona, Ruolo, SUM(Salario) AS Salario
3 FROM Lavoratore l
4 GROUP BY Id_Persona, Ruolo) AS sq JOIN Persona p ON (p.Id=sq.Id_Persona)
5 GROUP BY sq.Ruolo
6 --Risultato Query
7 +-----+-----+-----+-----+-----+-----+
8 | ID | Nome      | Cognome | Sesso | Ruolo          | Salario |
9 +-----+-----+-----+-----+-----+-----+
10 | 1 | Mario     | Rossi   | M     | Attore         | 25000 |
11 | 4 | Alessandro | Rebeschi | M     | Regista        | 30000 |
12 | 8 | Enrico    | Marcato | M     | Produttore     | 50000 |
13 | 4 | Alessandro | Rebeschi | M     | Sceneggiatore  | 15000 |
14 +-----+-----+-----+-----+-----+-----+
```

### Query5 Lista film a cui ha partecipato uno specifico attore.

In questo caso viene preso come esempio la persona con Id=21.

```
1 SELECT f.Titolo, f.Anno, f.Durata, f.Genere
2 FROM Film f JOIN `Cast` c ON c.Id=f.Id_Cast JOIN Lavoratore l ON l.Id_Cast=c.Id
3 WHERE l.Ruolo='Attore' AND l.Id_Persona=21
4 --Risultato Query
5 +-----+-----+-----+-----+
6 | Titolo          | Anno | Durata | Genere |
7 +-----+-----+-----+-----+
8 | Film Indie      | 2014 | 00:50:00 | Thriller |
9 | Storie Spaventose | 2010 | 00:50:00 | Horror   |
10 +-----+-----+-----+-----+
```

### Query6 Lista rapporto incasso a spettacolo

```
1 SELECT s.Id AS Spettacolo, s.Nome_Cinema AS Cinema, f.Titolo AS Titolo, COUNT(b.
   Id_Persona) AS Affluenza, COUNT(b.Id_Persona)*10 as Incasso
2 FROM Spettacolo s LEFT JOIN Biglietto b ON b.Id_Spettacolo=s.Id JOIN Film f ON f.Id=s.
   Id_Film
3 GROUP BY s.Id
4 ORDER BY Affluenza DESC
5 --Risultato Query
6 +-----+-----+-----+-----+-----+
7 | Spettacolo | Cinema      | Titolo          | Affluenza | Incasso |
8 +-----+-----+-----+-----+-----+
9 | 11 | D4C Marcon   | The Matrix      | 7         | 70      |
10 | 21 | D4C Silea    | The Matrix      | 3         | 30      |
11 | 1  | D4C Mestre   | The Matrix      | 2         | 20      |
12 | 41 | D4C Treviso  | The Matrix      | 2         | 20      |
13 | ... | ...         | ...            | ...       | ...     |
14 +-----+-----+-----+-----+-----+
```

## 5.2 Indici

### 5.2.1 Studio caso d'uso

Ipotizzando di avere il database con un'ampia quantita' di dati, si ipotizza che:

- La tabella **Spettacolo** contenga molte tuple
- Nella tabella **Spettacolo** vengono aggiunti molto frequentemente nuove tuple
- La stessa viene consultata spesso per l'orario inizio spettacoli
- La tabella **Film** contenga molte tuple
- Nella tabella **Film** poche volte al mese vengano aggiunti nuovi film
- La stessa viene consultata spesso per il titolo del film
- La tabella **Persona** contenga molte tuple
- Nella tabella **Persona** vengono aggiunti spesso nuove tuple
- La stessa viene consultata spesso per la ricerca di un film tramite attore, ricercandolo per cognome

### 5.2.2 Riconoscimento tabelle da indicizzare

- **Spettacolo**
- **Film**
- **Persona**

### 5.2.3 Implementazione dell'indice

```
1 CREATE INDEX idx_OrarioSpettacoli ON Spettacolo (Dataora_inizio);
2 CREATE INDEX idx_TitoloFilm ON Film (Titolo);
3 CREATE INDEX idx_CognomePersona ON Persona (Cognome);
```