

# Set Cinematografico

---

```
mysqldump -u studente -p cinema > cinema.sql
```

## Istruzioni

---

Per poter utilizzare l'applicativo si deve disporre di un database denominato cinema in mysql, basta utilizzare la seguente query sql

```
CREATE DATABASE cinema;  
USE cinema;
```

Per poter funzionare l'applicativo utilizza gradle, quindi si usa il comando run da IDEA oppure eclipse, o **piu' comodamente da terminale** con il comando

```
./gradlew run
```

## Struttura del progetto

La presente documentazione tratta nel dettaglio la progettazione e l'implementazione dell'elaborato "Set cinematografico" di Michele Nardini() Santoro Matteo(0000881608) Manuel Luzietti()

### Struttura:

- [Set Cinematografico](#)
  - [Struttura del progetto](#)
  - [Struttura:](#)
  - [Introduzione](#)
- [2 Analisi requisiti](#)
  - [2.11 Intervista](#)
- [2.2 Estrazione Concetti Fondamentali \(va cambiato se cambiamo il paragrafo sopra\)](#)
  - [3 Progetto dello schema concettuale](#)
  - [3.1 Distribuzione e Incasso](#)
  - [3.2 Ispirazione e Sceneggiatura](#)
  - [3.3 Gestione Fondi](#)
  - [3.4 Membro della Troupe](#)
  - [3.5 Stipendio](#)
  - [3.6 Scena](#)
  - [Schema Completo](#)
  - [4 Il progetto Logico](#)

- 4.1 Traduzione delle operazioni in query
  - 4.11 Aggiunta Film
  - 4.12 Aggiunta Membro Troupe
  - 4.13 Distribuzione
  - 4.14 Costumi e Magazzini
  - 4.15 Sponsor, Finanziatori e Fondo
  - 4.16 Retribuzione Membro\_Troupe
- 5 Specifiche funzionali
  - 5.01 Stipendio membri della troupe
  - 5.02 Elenco oggetti acquistati in magazzino
  - 5.03 Profitto finanziatori
  - 5.04 Costumi da usare per scena
  - 5.05 Dipendenti in scena
  - 5.06 Oggetti in scena
  - 5.07 Stipendio netto dipendente
  - 5.08 Profitto Produttori
  - 5.09 Costo affitto location per film
  - 5.10 Costo stipendi troupe per film
  - 5.11 Guadagno finanziatori
  - 5.12 Fatturato
  - 5.13 Ricavo
- 6 Il Progetto Logico
  - 6.1 Frequenza e costo degli accessi
  - 6.2 Volume dati del database
  - 6.3 Tabella degli accessi
  - 6.4 Traduzione delle entità
  - 6.5 Creazione delle tables
- 7 Interfaccia grafica
  - 7.1 Descrizione dell'architettura utilizzata
  - 7.2 Interfaccia di inserimento
  - 7.3 Interfaccia di visualizzazione
  - 7.4 Interfaccia per operazioni specifiche
- # Special thanks

## Introduzione

Il gruppo si pone come obiettivo quello di realizzare un database per la gestione di un set cinematografico coprendone tutti gli aspetti.

Saranno memorizzate all'interno del database le varie figure che partecipano alla realizzazione di un girato, gli enti che si occuperanno della distribuzione la gestione delle scene e dei ciak quindi gli stipendi e gli incassi.

## 2 Analisi requisiti

---

### 2.11 Intervista

Netflix Italia possiede un sistema informatico per la gestione del personale e delle risorse coinvolte della realizzazione di serie Tv e film che necessita di essere riadattato alle esigenze attuali. Si richiede quindi la progettazione di una database che permetta di gestire gli aspetti fondamentali nella realizzazione di uno sceneggiato. Il database sarà utilizzato da persone che si occuperanno della gestione. Saranno memorizzate tutte le informazioni relative ai membri della troupe che lavorano alla realizzazione di un film (fulcro attorno al quale gira l'applicativo), di cui memorizzeremo le principali informazioni ed eventuali serie letterarie da cui è tratto. Si vuole memorizzare informazioni di enti che si occuperanno della distribuzione e gli incassi che verranno generati. Come accade spesso in questo ambiente, per poter esistere alcuni film vengono stanziati dei finanziamenti che possono provenire da sponsor o finanziatori, ossia persone che possiedono un enorme capitale e che investono nella riuscita di un film, dal quale poi ricaveranno un guadagno. Il guadagno è calcolato in percentuale al guadagno totale del film. Le figure professionali che ruotano attorno la creazione di un girato riguardano ruoli come sceneggiatore, produttore (con annessa percentuale guadagno), produttore esecutivo, aiuto regista, capo regista (con annessa percentuale guadagno), regista, attore, stilista, operatore fonico e operatore fotografico di cui si vuole memorizzare l'anagrafica, telefono e IBAN che sarà poi utilizzato per l'accredito della paga mensile calcolata. Il sistema dovrà anche poter contenere uno storico delle buste paga. Si vogliono anche mantenere gli indirizzi di residenza citati nel sistema in maniera indipendente. Gli attori utilizzeranno costumi progettati da stilisti che vengono conservati in magazzini assieme agli oggetti di scena. Saranno infine registrati nell'applicativo i ciak presi durante le riprese del film, nello specifico il rullo su cui è impresso, il numero del ciak, la location nel quale viene girato e i costumi e oggetti di scena presenti nel ciak.

## 2.2 Estrazione Concetti Fondamentali

---

Netflix Italia possiede un sistema informatico per la gestione del personale e delle risorse coinvolte della realizzazione di serie Tv e **film** che necessita di essere riadattato alle esigenze attuali. Si richiede quindi la progettazione di una database che permetta di gestire gli aspetti fondamentali nella realizzazione di uno sceneggiato. Il database sarà utilizzato da persone che si occuperanno della gestione. Saranno memorizzate tutte le informazioni relative ai **membri della troupe** che lavorano alla realizzazione di un film (fulcro attorno al quale gira l'applicativo), di cui memorizzeremo le principali informazioni ed eventuali **serie letterarie** da cui è tratto. Si vuole memorizzare informazioni di **enti** che si occuperanno della **distribuzione** e gli **incassi** che verranno generati. Come accade spesso in questo ambiente, per poter esistere alcuni film vengono stanziati dei **finanziamenti** che possono provenire da **sponsor** o **finanziatori**, ossia persone che possiedono un enorme capitale e che investono nella riuscita di un film, dal quale poi ricaveranno un guadagno. Il **guadagno è calcolato in percentuale al guadagno totale** del film. Le figure professionali che ruotano attorno la creazione di un girato riguardano ruoli come **sceneggiatore, produttore (con annessa percentuale guadagno), produttore esecutivo, aiuto regista, capo regista (con annessa percentuale guadagno), regista, attore, stilista, operatore fonico e operatore**

**fotografico** di cui si vuole

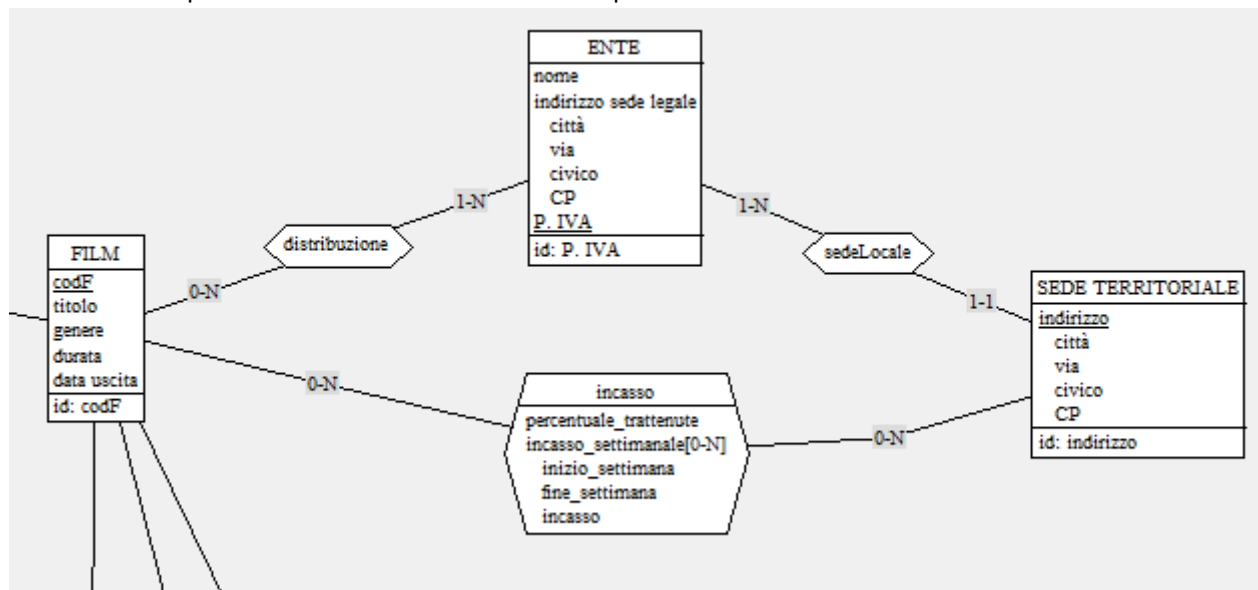
memorizzare l'anagrafica, telefono e IBAN che sarà poi utilizzato per l'accredito della **paga mensile** calcolata. Il sistema dovrà anche poter contenere uno storico delle **buste paga**. Si vogliono anche mantenere gli **indirizzi** di residenza citati nel sistema in maniera indipendente. Gli attori utilizzeranno **costumi** progettati da **stilisti** che vengono conservati in **magazzini** assieme agli **oggetti di scena**. Saranno infine registrati nell'applicativo i **ciak presi** durante le riprese del film, nello specifico il rullo su cui è impresso, il numero del ciak, la location nel quale viene girato e i costumi e oggetti di scena presenti nel ciak.

### 3 Progetto dello schema concettuale

Lo schema concettuale è stato realizzato attorno ad alcune entità di base come i membri della troupe dai quali deriviamo i vari lavori e maestranze e l'entità scena-Ciak fondamentale per la realizzazione di un film

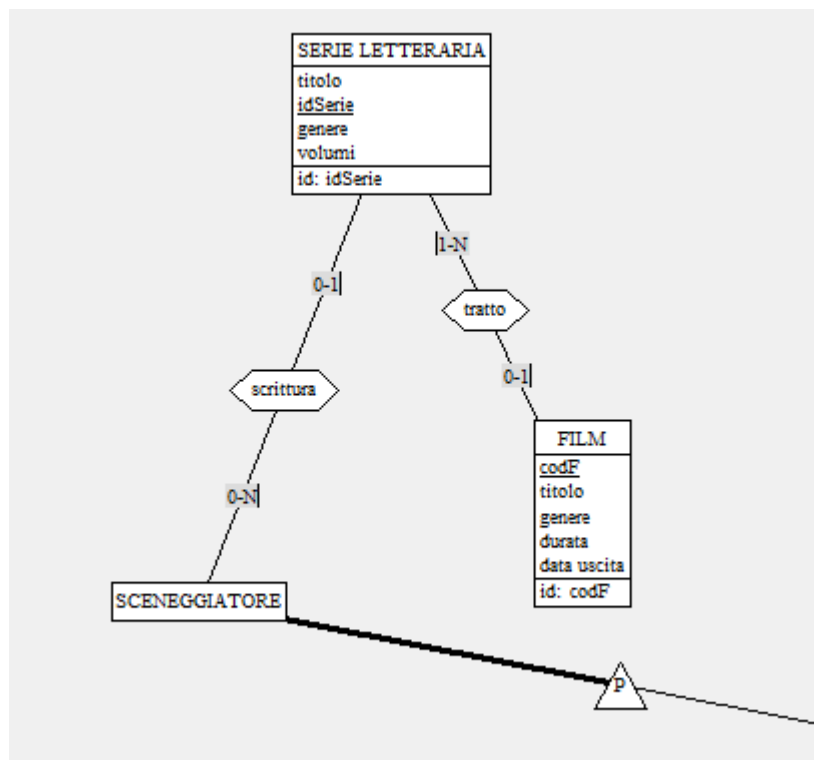
#### 3.1 Distribuzione e Incasso

Ogni Film ha bisogno di un ente specifico che si occupi della distribuzione, come ad esempio l'azienda UCI che dopo aver comprato i diritti per la riproduzione porta nelle sue sedi la pellicola, ogni ente da noi gestito attraverso un entità composta da nome, indirizzo, P.IVA ha poi una dislocazione locale.



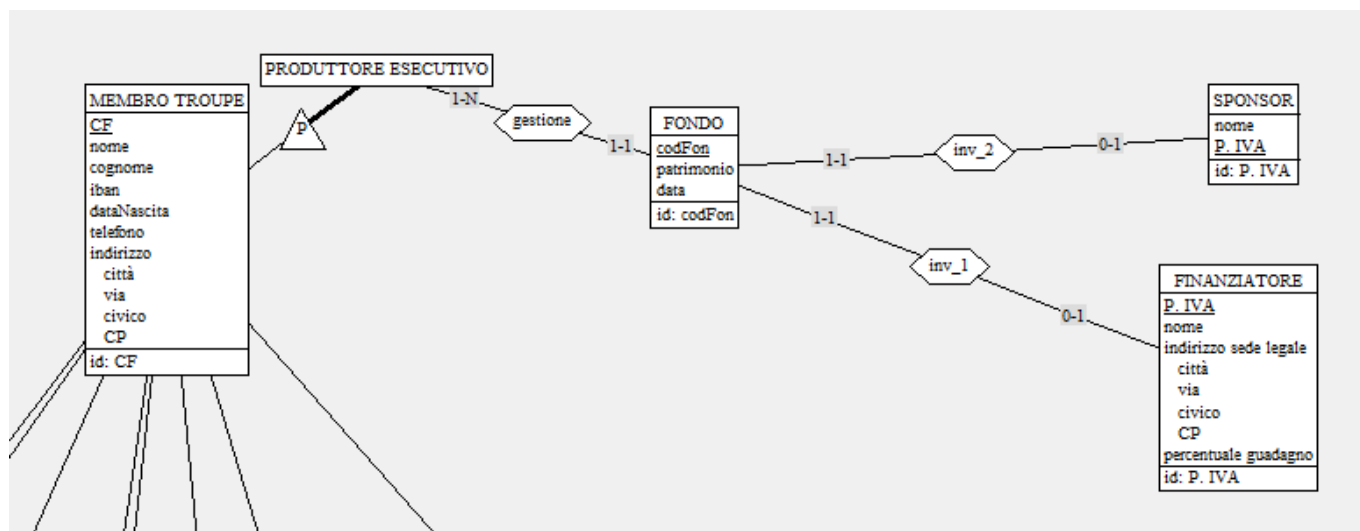
#### 3.2 Ispirazione e Sceneggiatura

E' comune che i film siano stati ispirati da alcune serie letterarie, come ad esempio la saga di Harry Potter, abbiamo quindi deciso d'inserire la possibilità che un dato film sia tratto da una Serie Letteraria che abbiamo modellato come un entità, spesso capita nell' ambiente cinematografico che lo Sceneggiatore sia anche l'autore della Serie



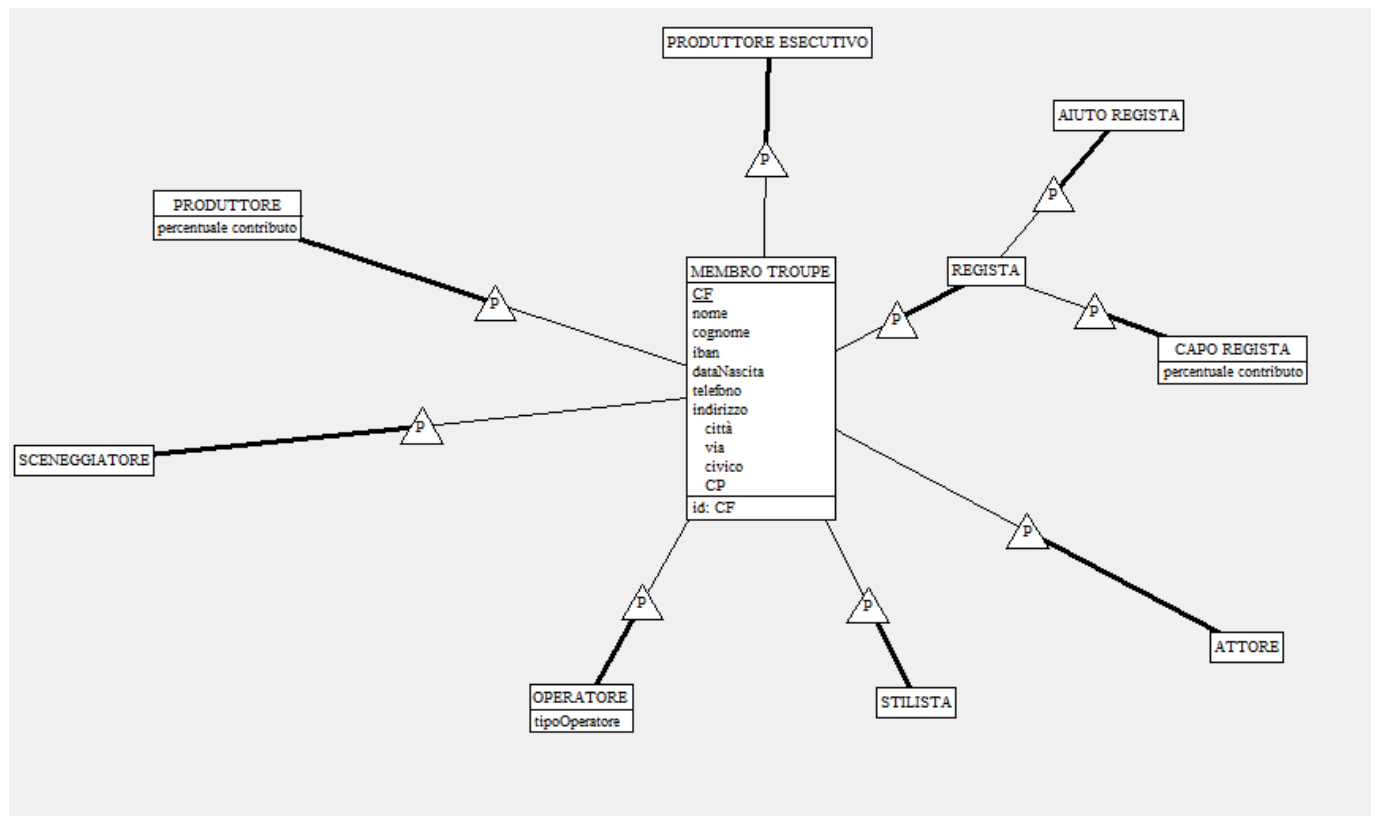
### 3.3 Gestione Fondi

Un film per poter essere realizzato ha un enorme bisogno di soldi quindi di fondi dato che i costi sono molti, spesso questo patrimonio si crea dagli investimenti che la pellicola riesce a raccogliere, abbiamo creato le entità Sponsor che rappresentano le possibili aziende che possono in cambio di una sponsorizzazione del loro prodotto pagare una somma di denaro e l'entità Finanziatore per la persona o azienda che decidono d'investire una somma di denaro per poi poter guadagnare grazie a una percentuale una volta ricevuti gli incassi del film tramite un'associazione che ha come attributo la data d'investimento e il Produttore esecutivo abbiamo modellato la gestione dei fondi. A livello concettuale avremo una gerarchia con entità padre il fondo ed entità figlie i corrispettivi fondo\_sponsor e fondo\_finanziatore che ereditano gli attributi di fondo con la chiave esterna di sponsor o finanziatore.



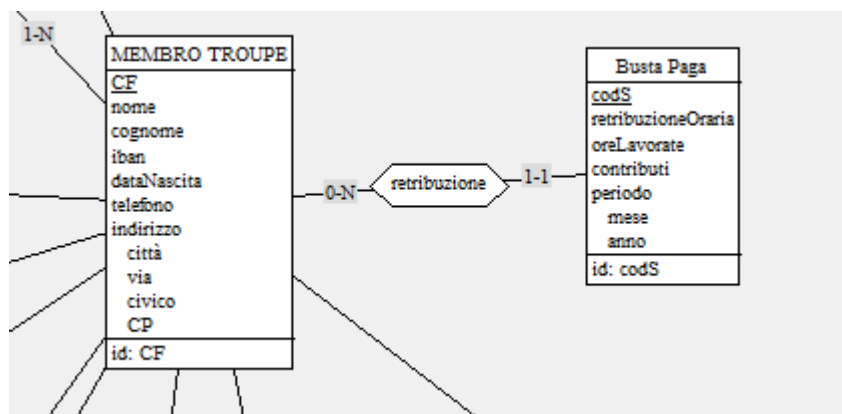
### 3.4 Membro della Troupe

In un set cinematografico le persone che lavorano alla realizzazione di un film sono tante e compiono lavori diversi, ma abbiamo deciso di utilizzare una gerarchia per poter più comodamente rappresentarle.

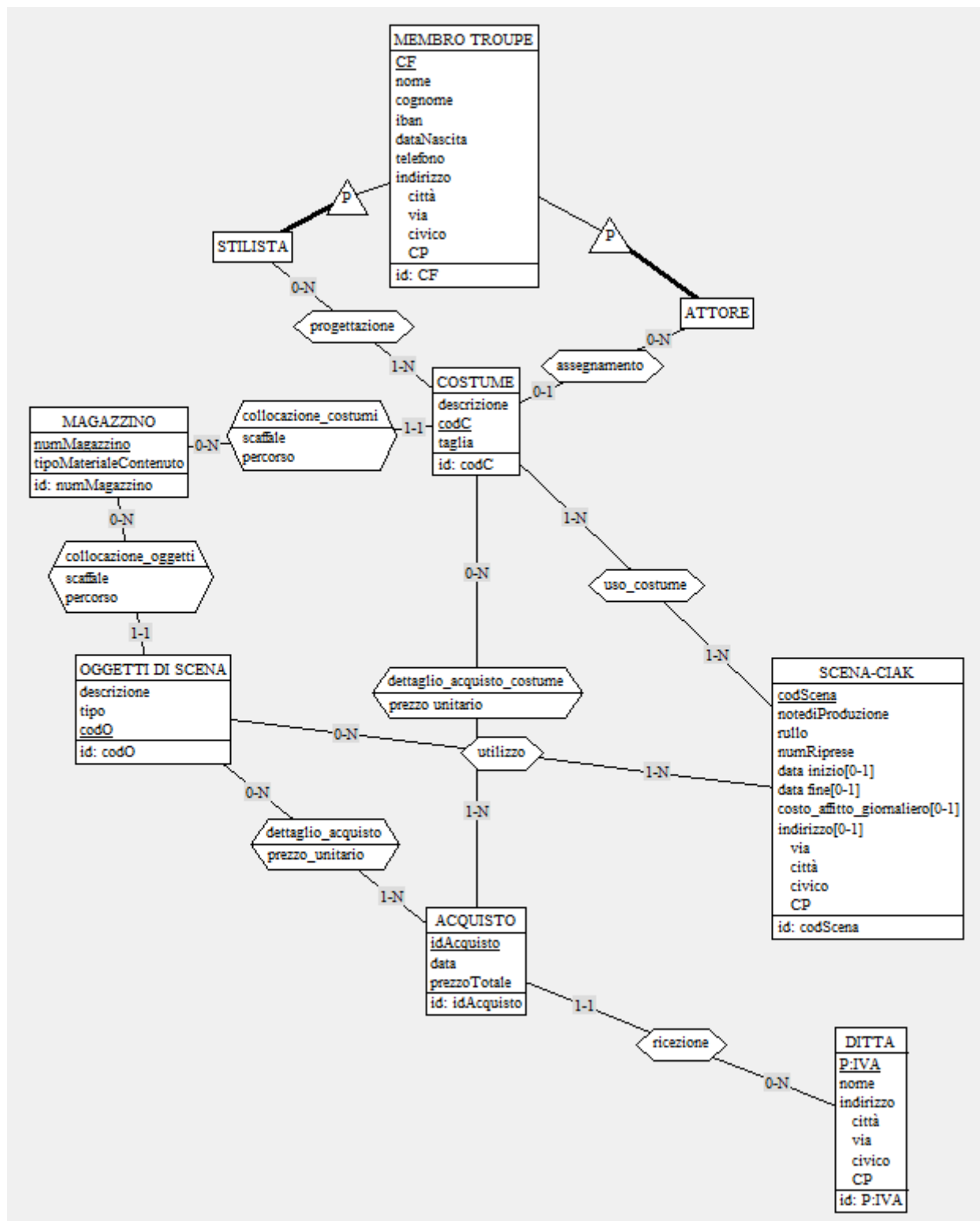


### 3.5 Stipendio

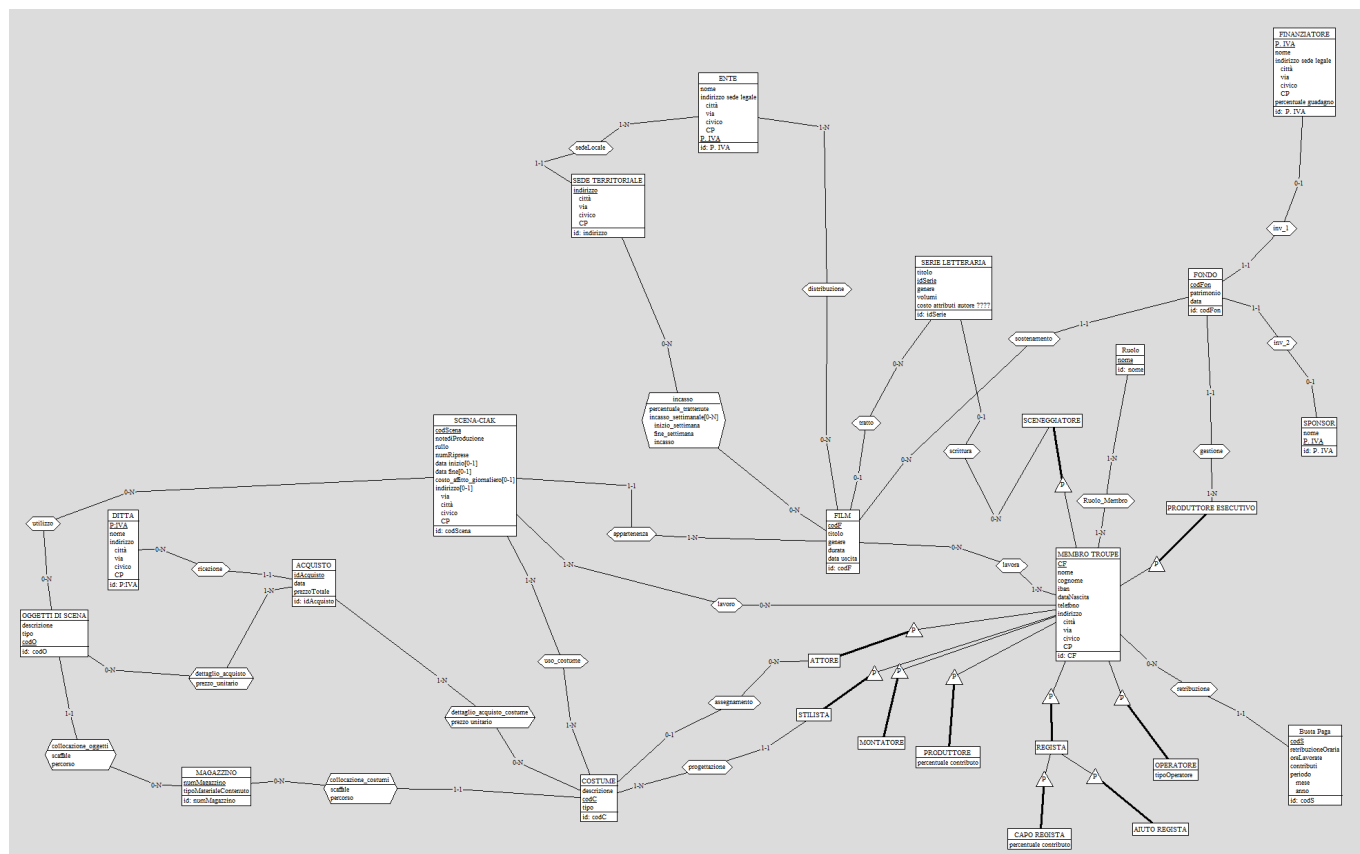
Ogni addetto ai lavori in un set cinematografico ha uno stipendio, modellato attraverso un'entità nella quale vengono registrati i codici singoli per busta paga, le ore accumulate sul set, i contributi e il periodo sul quale poi verrà calcolato un vero e proprio stipendio.



### 3.6 Scena



## Schema Completo



Qui riportati gli schemi di navigazione delle operazioni

## 4.1 Traduzione delle operazioni in query

Aggiungere un film consiste nell'aggiungere un'istanza dell' entità FILM

```
INSERT IGNORE INTO Film(codF, titolo, genere, durata, dataUscita, idSerie)
VALUES (00001, 'Star Wars: Episodio VI - Il ritorno dello Jedi',
        'fantascienza', 134, 25/05/2983, 2551983);
```

Aggiungiamo prima al database un membro generico, il quale verra' poi collegato al film tramite la tupla Film\_Membro\_Troupe che rappresenta l'associazione *lavora* e ad il lavoro che svolgerà all' interno di questo progetto con una tupla della tabella RuoloMembroTroupe

```
INSERT IGNORE INTO MembroTroupe(CF, nome, cognome, iban, dataNascita, telefono,
                                codInd, percentualeContributo)
-- produttore esecutivo George Lucas
```



```
VALUES ('GRGLCS14ES44', 'George', 'Lucas', 'GB98BARC20040156884556',
        '1944/05/14', '516-527-8719', 24673, 1.0);

INSERT IGNORE INTO RuoloMembroTroupe(CF, nomeRuolo)
VALUES ('GRGLCS14ES44', 'Produttore Esecutivo');

INSERT IGNORE INTO Film_Membro_Troupe(codF, CF)
-- George lucas
VALUES (1, 'GRGLCS14ES44'),
```

### 4.13 Distribuzione

Avendo precedentemente inserito un Indirizzo e usando quel codice, creiamo l'ente che si occuperà della distribuzione del film, rappresentando quindi l'associazione e l'entità ente con la sua eventuale sedeTerritoriale.

```
INSERT IGNORE INTO Enti(P_IVA,nome,codInd)
-- ente di distribuizione
VALUES (40365320379, "UCI Milano", 18302);

INSERT IGNORE INTO Indirizzo(codInd, citta, via, civico, CAP)
--sede territoriale uci savignano
VALUES (16395, 'savignano', 'Piazza Metropolis', 18, 47039),

INSERT IGNORE INTO SediTerritoriali(P_IVA, codInd)
VALUES (40365320379, 16395);
```

### 4.14 Costumi e Magazzini

Per poter procedere a realizzare i concetti dell'utilizzo e storage dei costumi si può inserire l'entità Magazzino, le associazioni riportate nell'ER collocazioni\_costumi e assegnamento attore vengono formalizzate utilizzando un'Entità Posizione Magazzino che utilizza la primary key di Magazzino che viene legata a Costume con la foreign key codP e la seconda associazione assegnando a un costume una foreign key CF (codice fiscale) che viene importato da MembroTroupe.

```
INSERT IGNORE INTO Magazzino(numMagazzino, codInd)
-- magazzino principale
VALUES(1,19341);

INSERT IGNORE INTO Costume(codC, tipo, descrizione, CF, codP)
-- costume Luke
VALUES(49262, 'fantasia', 'Costume di Luke Skywalker, kimono nero,
        stivali, spada Laser Verde', 'MRKHML25IS51', 25588);

INSERT IGNORE INTO PosizioneMagazzino(codP,numMagazzino,scaffale, percorso)
-- Posizione costume luke skywalker
VALUES(25588, 1, 2, 'S');
```

## 4.15 Sponsor, Finanziatori e Fondo

Per poter procedere alla realizzazione di un film, quest'ultimo avrà bisogno di una serie di fondi che andranno a contribuire alle spese. Per fare ciò inseriremo prima una serie di Sponsor e Finanziatori, seguiti da una entità fondo aventi due foreign key associate opzionalmente o allo sponsor o al finanziatore (ipotizzando che un fondo può essere o di uno sponsor o di un finanziatore).

```
INSERT IGNORE INTO Sponsor(P_IVA_SPONSOR, nome)
-- Samsung
VALUES('53179880082', 'Samsung Galaxy');

INSERT IGNORE INTO Finanziatore(P_IVA_FINANZIATORE, nome, codInd
, percentualeGuadagno)
-- Finanziatore 1
VALUES('31562270996', 'Micheal Barnaby', 49429, 1.8);

INSERT IGNORE INTO Fondo(codFondo, dataAccredito, patrimonio, P_IVA_SPONSOR,
P_IVA_FINANZIATORE, codF)
-- george lucas ha contribuito al fondo
VALUES (1, '1977-01-05', 2000000, '53179880082', null, 1);
```

## 4.16 Retribuzione Membro Troupe

Avendo creato in precedenza un membro troupe, ora potremo creare le varie busta paga per quel determinato membro troupe usando una entità busta paga contenente tutte le informazioni dello stipendio del dipendente e una entità retribuzione che avrà lo scopo di associare un membro troupe alla propria busta paga

```
INSERT IGNORE INTO MembroTroupe(CF, nome, cognome, iban, dataNascita, telefono,
                                codInd, percentualeContributo)
-- produttore esecutivo George Lucas
VALUES ('MRKHML25IS51', 'Mark', 'Hamill', 'GB56BARC20038438638758',
        '1951/09/25', '214-989-5138', 37065, NULL);

INSERT IGNORE INTO BustaPaga(codB, retribuzioneOraria, oreLavorate, mese)
-- luke skywalker
VALUES (47918, 32, 111.2, 'aprile');

INSERT IGNORE INTO Retribuzione(CF, CodB)
-- luke
VALUES ('MRKHML25IS51', 47918);
```

# 5 Specifiche funzionali

---

Ora andremo a svolgere le seguenti operazioni

## 5.01 Stipendio membri della troupe

Query creata per poter calcolare lo stipendio della troupe per un mese, ottenuta collezionando le busta paga dei lavoratori, in altre parole il costo della troupe mensile, il tutto ottenuto selezionando il Film per il quale si desidera conoscere la cifra.

```
select @stipendi := sum(retribuzioneOraria * oreLavorate) as
CostoTroupe_Mese from
BustaPaga b join Retribuzione r on (b.codB = r.codB )
join Film_Membro_Troupe flm on (r.CF = flm.CF) where mese = ?
and codF = ?;
```

## 5.02 Elenco oggetti acquistati in magazzino

---

Trova la posizione in un dato magazzino di tutti gli oggetti acquistati specificandone magazzino.

```
select ods.tipo, ods.descrizione, pm.scaffale, pm.percorso
from PosizioneMagazzino pm, OggettiDiScena ods
where pm.codP = ods.codP
and pm.numMagazzino = ?;
```

## 5.03 Profitto finanziatori

---

Query creata per poter calcolare il profitto ottenuto dai finanziatori grazie al finanziamento di uno specifico film. Tale profitto è calcolato anche in base alla percentuale di guadagno stabilita.

```
select @Denaro := sum(incasso) as money FROM Incasso;
select distinct F.nome, F.percentualeGuadagno, (F.percentualeGuadagno / 100 *
@Denaro ) as guadagno
from Finanziatore F join Fondo ff on (F.P_IVA_FINANZIATORE =
ff.P_IVA_FINANZIATORE)
where F.percentualeGuadagno is not null
and codF = ?
order by guadagno DESC;
```

## 5.04 Costumi da usare per scena

---

Query creata per definire i costumi che saranno da usare nella realizzazione di una determinate scena e che dovrà indossare un attore X

```
select c.descrizione, c.tipo
from ScenaCiak sc join CostumeScena cs on (cs.codScena=sc.codScena)
```

```
join Costume c on (c.codC = cs.codC)
where sc.noteDiProduzione = ?;
```

## 5.05 Dipendenti in scena

---

```
select mt.*
from ScenaCiak sc join MembroTroupeScena mts on (sc.codScena = mts.codScena)
join MemtroTroupe mt on (mts.CF = mt.CF)
where sc.codScena = ?;
```

## 5.06 Oggetti in scena

---

```
select ods.*
from ScenaCiak sc join OggettoScena os on (sc.codScena=os.codScena)
join OggettiDiScena ods on (os.cod0=ods.cod0)
where sc.codScena=?;
```

## 5.07 Stipendio netto dipendente

---

Definisce lo stipendio di un determinato membro della troupe in un dato mese, calcolandolo moltiplicando la retribuzione oraria stabilita mediante contratto e le ore lavorate

```
select @stipendio := sum(retribuzioneOraria * oreLavorate) as Stipendio
from BustaPaga bp join Retribuzione r on (bp.codB = r.codB)
where r.CF = ?;
```

## 5.08 Profitto produttori

---

Serve per poter calcolare quanti soldi riescono a ricavare dalla produzione di un film i produttori e chiunque lavori all'interno e riceva una percentuale dagli incassi.

```
select @Denaro := sum(incasso) as money FROM Incasso;
select distinct M.nome, M.cognome,M.percentualeContributo,
(M.percentualeContributo / 100 * @Denaro ) as guadagno, Rm.nomeRuolo
from Incasso I, MembroTroupe M, RuoloMembroTroupe Rm join Film_Membro_Troupe flm
on (Rm.CF = flm.CF)
where (M.CF = Rm.CF) and flm.codF = ? and M.percentualeContributo is not null
order by guadagno DESC;
```

---

## 5.09 Costo affitto location per film

---

Costo totale di affitto delle location per girare un determinato film

```
select @Affitto := sum(costoAffittoGiornaliero * durataOre) as affittoOrario
from ScenaCiak where codF = codiceFilm;
```

---

## 5.10 Costo stipendi troupe per film

---

Calcolo del costo degli stipendi per ogni membro che ha lavorato alla realizzazione di un film

```
select @StipendiTot := sum(retribuzioneOraria * oreLavorate) as
stipendio_membro from
BustaPaga b join Retribuzione r on (b.codB = r.codB )
join Film_Membro_Troupe flm on (r.CF = flm.CF)
and codF = codiceFilm;
```

---

## 5.11 Guadagno finanziatori

---

Guadagno totale dei finanziatori per l'investimento in un film

```
select @Guadagno := sum(F.percentualeGuadagno / 100 * @fatturato ) as
guadFinanziatore
from Finanziatore F join Fondo ff on (F.P_IVA_FINANZIATORE =
ff.P_IVA_FINANZIATORE)
where F.percentualeGuadagno is not null
and codF = codiceFilm;
```

---

## 5.12 Fatturato

---

Fatturato totale del film

```
select @fatturato := sum(incasso) as Fatturato from Incasso
where codF = codiceFilm;
```

---

## 5.13 Ricavo

---

Ricavo totale di un determinato film -> Ricavo = Guadagno - Spese

```
select (
  (select @fatturato := sum(incasso) as Fatturato from Incasso
  where codF = ?) -
  (SELECT SUM(ricavo) ricavo
  from
  (
    select sum(costoAffittoGiornaliero * durataOre) ricavo from ScenaCiak
  where codF = ?
  UNION
  select sum(retribuzioneOraria * oreLavorate) ricavo from
  BustaPaga b join Retribuzione r on (b.codB = r.codB )
  join Film_Membro_Troupe flm on (r.CF = flm.CF)
  and codF = ?
  UNION
  select sum(F.percentualeGuadagno / 100 * @fatturato ) ricavo
  from Finanziatore F join Fondo ff on (F.P_IVA_FINANZIATORE =
  ff.P_IVA_FINANZIATORE)
  where F.percentualeGuadagno is not null
  and codF = ?
  ) s)
) as ricavo;
```

## 6 Il Progetto Logico

---

### 6.1 Frequenza e costo degli accessi

---

Numero	Descrizione	Frequenza
1	Aggiunta film	20/mese
2	Aggiunta membro troupe	1000/mese
3	Aggiungere una nuova sede territoriale per un dato ente	5/anno
4	Aggiunta di un nuovo ente	1/anno
5	Aggiunta di un fondo proveniente da uno sponsor/finanziatore	100/mese
6	Aggiunta busta paga per un dato membro di un film	20/mese
7	Calcolo stipendio netto della troupe in un determinato mese	1000/mese
8	Elenco degli oggetti in magazzino	5/mese
9	Profitto finanziatori	10/mese
10	Luogo di ripresa	5/giorno
11	Costumi da usare per scena	5/giorno
12	Dipendenti in scena	5/giorno
13	Oggetti in scena	5/giorno
14	Calcolo stipendio netto di un membro della troupe in un determinato mese	1000/mese
15	Profitto Produttori	1/settimana
16	Costo affitto location per film	1/giorno
17	Costo stipendi per film	1/mese
18	Guadagno finanziatori per film	1/settimana
19	Fatturato	1/giorno
20	Ricavi	1/giorno

## 6.2 Volume dati del database

---

Riportiamo in seguito la tabella dei volumi in cui sono elencate tutte le entità, le relazioni e i relativi volumi.

Nome	Tipo	Volume
Serie Letteraria	E	300
Film	E	500
Indirizzo	E	2000
Ente	E	10
Sede Territoriale	E	500
Fondo	E	5000
Sponsor	E	100
Finanziatore	E	100
Membro Troupe	E	5000
Ruolo	E	10
Busta Paga	E	30000
Scena Ciak	E	10000
Magazzino	E	1
Posizione Magazzino	E	500
Costume	E	1000
Oggetto di Scena	E	1000
Acquisto	E	1000
Ditta	E	50
Distribuzione	R	5000
Incasso	R	10000
Ruolo Membro Troupe	R	6000
Retribuzione	R	30000
Film Membro Troupe	R	10000
Membro Troupe Scena	R	100000
Stilista Costume	R	500
Costume Scena	R	50000
Oggetto Scena	R	50000
Acquisto costume	R	500
Acquisto oggetto	R	1000

## 6.3 Tabella degli accessi

---



Operazione	Concetto	Costrutto	Accesso	Tipo
Aggiunta membro troupe	Membro Troupe	E	1	S
	Ruolo membro troupe	A	1	S
	Film membro troupe	A	1	S
Stipendio Membri Troupe	Busta paga	E	5000	L
	Retribuzione	A	5000	L
	Film membro troupe	A	5000	L
Elenco oggetti acquistati in magazzino	Oggetto di Scena	E	1000	L
	Posizione Magazzino		1000	L
Profitto finanziatori	Film	E	1	L
	Incasso	A	20	L
	Fondo	A	10	L
	Finanziatore	A	10	L
Luoghi riprese	Scena ciak	E	1	L
	Indirizzo	E	1	L
Costumi da usare per scena	Scena ciak	E	1	L
	Costume scena	A	5	L
	Costume	E	5	L
	MembroTroupe Scena	A	5	L
	Membro Troupe	E	5	L
Dipendenti in scena	Scena ciak	E	1	L
	Membro Troupe	E	20	L
Oggetti in scena	Scena ciak	E	1	L
	Oggetto in scena	A	10	L
	Oggetto di scena	E	10	L
Stipendio netto dipendente	Membro troupe	E	1	L
	Busta paga	E	1	L
	Retribuzione	A	1	L
Profitto finanziatori	Incasso	A	20	L
	MembroTroupe	E	10	L
	Ruolo membro troupe	A	10	L
	Film membro troupe	A	10	L
Costo affitto location	Scena ciak	E	20	L
Costo stipendi per film	Busta paga	E	60	L
	Film membro troupe	A	60	L
Guadagno finanziatori per film	Finanziatore	E	10	L
	Fondo	A	10	L
Fatturato	Incasso	A	10	L
Ricavo film	Incasso	A	20	L
	MembroTroupe	E	10	L
	Ruolo membro troupe	A	10	L
	Film membro troupe	A	10	L
	Scena ciak	E	20	L
	Busta paga	E	60	L
	Film membro troupe	A	60	L
	Finanziatore	E	10	L
	Fondo	A	10	L
	Incasso	A	10	L

## 6.4 Traduzione delle entità

---

- Film(**codF**, titolo, genere, durata, dataUscita, idSerie[0-1])
  - FK: idSerie REFERENCES **Serie\_Letteraria**
- Indirizzo(**codInd**, città, via, civico, CAP)
- Ente(**P.IVA\_ENTE**, nome, codInd)
  - FK: codInd REFERENCES **Indirizzo**
- Distribuzione(**P.IVA**, **codF**)
  - FK: P.IVA\_ENTE REFERENCES **Ente**
  - FK: codF REFERENCES **Film**
- Sedi\_Territoriali(P\_IVA, **codInd**)
  - FK: codInd REFERENCES **Indirizzo**
- Incasso(dataInizio, **dataFine**, **incasso**, **codF**, codInd)
  - FK: codF REFERENCES **Film**
  - FK: codInd REFERENCES **Indirizzo**
- Fondo(**codFondo**, dataAccredito, patrimonio, P\_IVA\_SPONSOR[0, 1], P\_IVA\_FINANZIATORE[0, 1], codF)
  - FK: P.IVA\_SPONSOR REFERENCES **Sponsor**
  - FK: P.IVA\_FINANZIATORE REFERENCES **Finanziatore**
  - FK: codF REFERENCES **Film**
- Sponsor(**P.IVA\_SPONSOR**, nome)
- Finanziatore(**P.IVA\_FINANZIATORE**, nome, codInd, percentuale\_guadagno)
  - FK: codInd REFERENCES **Indirizzo**
- SerieLetteraria(**idSerie**, titolo, genere, CF)
  - FK: CF REFERENCES **Membro\_troupe**
- MembroTroupe(**CF**, nome, cognome, IBAN, dataNascita, telefono, codInd, percentualeContributo[0-1])
  - FK: codInd REFERENCES **Indirizzo**
- Ruolo(**nomeRuolo**)
- RuoloMembroTroupe(**CF**, **nomeRuolo**)
  - FK: CF REFERENCES **Membro\_troupe**
  - FK: nomeRuolo REFERENCES **Ruolo**
- BustaPaga(**codB**, retribuzioneOraria, oreLavorate, mese)

- Retribuzione(**CF**, **codB**)
  - FK: CF REFERENCES **Membro\_Troupe**
  - FK: codB REFERENCES **BustaPaga**
- FilmMembroTroupe(**codF**, **CF**)
  - FK: codF REFERENCES **Film**
  - FK: CF REFERENCES **Membro\_troupe**
- ScenaCiak(**codScena**, noteDiProduzione, rullo, numRiprese, durataOre, costoAffittoGiornaliero[0-1], codF)
  - FK: codF REFERENCES **Film**
- MembroTroupeScena(**codScena**, **CF**)
  - FK: codScena REFERENCES **Scena-Ciak**
  - FK: CF REFERENCES **Membro-Troupe**
- Magazzino(**numMagazzino**, codInd)
  - FK: codInd REFERENCES **Indirizzo**
- PosizioneMagazzino(**codP**, **numMagazzino**, scaffale, percorso)
  - FK: numMagazzino REFERENCES **Magazzino**
- Costume(**codC**, tipo, descrizione, CF, codP)
  - FK: CF REFERENCES **Membro\_troupe**
  - FK: codP REFERENCES **PosizioneMagazzino**
- StilistaCostume(**CF**, **codC**)
  - FK: CF REFERENCES **Membro\_troupe**
  - FK: codC REFERENCES **Costume**
- CostumeScena(**codC**, **codScena**)
  - FK: codC REFERENCES **Costume**
  - FK: codScena REFERENCES **Scena-Ciak**
- OggettiDiScena(**codO**, tipo, descrizione, codP)
  - FK: numMagazzino REFERENCES **PosizioneMagazzino**
- OggettoScena(**codO**, **codScena**)
  - FK: codO REFERENCES **Oggetto\_di\_scena**
  - FK: codScena REFERENCES **ScenaCiak**
- Acquisto(**idAcquisto**, data, prezzoTotale, P.IVA\_DITTA)
  - FK: P.IVA\_DITTA REFERENCES **Ditta**

- AcquistoCostume(**codC**, **idAcquisto**, prezzo)
  - FK: codC REFERENCES **Costume**
  - FK: idAcquisto REFERENCES **Acquisto**
- AcquistoOggetto(**codO**, **idAcquisto**, prezzo)
  - FK: codO REFERENCES **Oggetto\_di\_scena**
  - FK: idAcquisto REFERENCES **Acquisto**
- Ditta(**P.IVA\_DITTA**, nome, codInd)
  - FK: codInd REFERENCES **Indirizzo**

## 6.5 Creazione delle tables

---

```
CREATE TABLE if not exists Indirizzo(  
    codInd int primary key check(length(codInd) <= 5),  
    citta varchar(41) NOT NULL,  
    via varchar(40) NOT NULL,  
    civico int NOT NULL,  
    CAP int NOT NULL check(length(CAP) = 5)  
);  
  
CREATE TABLE IF NOT EXISTS Enti(  
    P_IVA varchar(31) NOT NULL,  
    nome varchar(40) NOT NULL,  
    codInd INT NOT NULL,  
    FOREIGN KEY (codInd) REFERENCES Indirizzo(codInd)  
    ON DELETE CASCADE  
    ON UPDATE NO ACTION,  
    PRIMARY KEY(P_IVA)  
);  
  
CREATE TABLE if not exists MembroTroupe(  
    CF varchar(18) primary key,  
    nome varchar(20) NOT NULL,  
    cognome varchar(20) NOT NULL,  
    IBAN varchar(30) NOT NULL,  
    dataNascita date NOT NULL,  
    telefono varchar(35) NOT NULL,  
    codInd INT NOT NULL,  
    FOREIGN KEY (codInd) REFERENCES Indirizzo(codInd)  
    ON DELETE CASCADE  
    ON UPDATE NO ACTION,  
    percentualeContributo float(3) check(percentualeContributo between 0 and 100)  
);  
  
CREATE TABLE if not exists BustaPaga(  
    codB INT,  
    retribuzioneOraria float NOT NULL,
```

```
    oreLavorate float NOT NULL,
    mese varchar(30) CHECK (mese in ('gennaio', 'febbraio', 'marzo', 'aprile',
'maggio', 'giugno', 'luglio', 'agosto', 'settembre', 'ottobre', 'novembre',
'dicembre')),
    primary key(codB)
);

CREATE TABLE if not exists Retribuzione(
    CF varchar(18),
    FOREIGN KEY (CF) REFERENCES MembroTroupe(CF)
    ON DELETE CASCADE
    ON UPDATE NO ACTION,
    codB INT,
    FOREIGN KEY (codB) REFERENCES BustaPaga(codB)
    ON DELETE CASCADE
    ON UPDATE NO ACTION,
    PRIMARY KEY(CF, codB)
);

CREATE TABLE IF NOT EXISTS Ruolo(
    nomeRuolo varchar(41) Primary KEY
);

CREATE TABLE IF NOT EXISTS RuoloMembroTroupe(
    CF varchar(18),
    FOREIGN KEY (CF) REFERENCES MembroTroupe(CF)
        ON DELETE CASCADE
        ON UPDATE NO ACTION,
    nomeRuolo varchar(41),
    FOREIGN KEY (nomeRuolo) REFERENCES Ruolo(nomeRuolo)
        ON DELETE CASCADE
        ON UPDATE NO ACTION,
    PRIMARY KEY(CF, nomeRuolo)
);

CREATE TABLE IF NOT EXISTS SerieLetteraria(
    idSerie varchar(11) primary key,
    titolo varchar(51) NOT NULL,
    genere varchar(40) NOT NULL,
    CF varchar(18) NOT NULL,
    FOREIGN KEY (CF) REFERENCES MembroTroupe(CF)
    ON DELETE CASCADE
    ON UPDATE NO ACTION
);

CREATE TABLE IF NOT EXISTS Film (
    codF int NOT NULL AUTO_INCREMENT,
    titolo varchar(51) NOT NULL UNIQUE,
    genere varchar(40) NOT NULL CHECK (genere = 'Animazione' or genere = 'Avventura'
or genere = 'Azione' or genere = 'Biografico' or genere = 'Commedia' or genere =
'Documentario' or genere = 'Drammatico' or genere = 'Pornografico' or genere =
'Fantascienza' or genere = 'Fantasy' or genere = 'Guerra' or genere = 'Horror' or
genere = 'Musical' or genere = 'Storico' or genere = 'Thriller' or genere =
'Western'),
```

```
durata int NOT NULL CHECK (durata > 0),
dataUscita date DEFAULT NULL,
idSerie varchar(11),
FOREIGN KEY (idSerie) REFERENCES SerieLetteraria(idSerie)
ON DELETE CASCADE
ON UPDATE NO ACTION,
PRIMARY KEY (codF)
);

CREATE TABLE IF NOT EXISTS SediTerritoriali (
P_IVA varchar(31) NOT NULL,
FOREIGN KEY (P_IVA) REFERENCES Enti (P_IVA)
ON DELETE CASCADE
ON UPDATE NO ACTION,
codInd INT NOT NULL,
FOREIGN KEY (codInd) REFERENCES Indirizzo(codInd)
ON DELETE CASCADE
ON UPDATE NO ACTION,
PRIMARY KEY (codInd)
);

CREATE TABLE if not exists Distribuzione(
P_IVA varchar(31) ,
FOREIGN KEY (P_IVA) REFERENCES Enti(P_IVA)
ON DELETE CASCADE
ON UPDATE NO ACTION,
codF INT NOT NULL,
FOREIGN KEY (codF) REFERENCES Film(codF)
ON DELETE CASCADE
ON UPDATE NO ACTION,
PRIMARY KEY(P_IVA, codF)
);

CREATE TABLE if NOT EXISTS Incasso(
dataInizio date NOT NULL,
dataFine date NOT NULL,
incasso int(11),
codF INT NOT NULL,
FOREIGN KEY (codF) REFERENCES Film(codF)
ON DELETE CASCADE
ON UPDATE NO ACTION,
codInd INT NOT NULL,
FOREIGN KEY (codInd) REFERENCES SediTerritoriali(codInd)
ON DELETE CASCADE
ON UPDATE NO ACTION,
PRIMARY KEY(dataInizio, dataFine, incasso)
);

CREATE TABLE if not exists Sponsor(
P_IVA_SPONSOR varchar(31) primary key,
nome varchar(41) NOT NULL
);
```

```
CREATE TABLE if not exists Finanziatore(  
    P_IVA_FINANZIATORE varchar(31) primary key,  
    nome varchar(41) NOT NULL,  
    codInd INT NOT NULL,  
    FOREIGN KEY (codInd) REFERENCES Indirizzo(codInd)  
        ON DELETE CASCADE  
        ON UPDATE NO ACTION,  
    percentualeGuadagno float(3) NOT NULL CHECK(percentualeGuadagno between 0 and  
100)  
);
```

```
CREATE TABLE if not exists Fondo(  
    codFondo INT primary key auto_increment,  
    dataAccredito date NOT NULL,  
    patrimonio float(14) NOT NULL CHECK(patrimonio >= 0),  
    P_IVA_SPONSOR varchar(31),  
    FOREIGN KEY (P_IVA_SPONSOR) REFERENCES Sponsor(P_IVA_SPONSOR)  
        ON DELETE CASCADE  
        ON UPDATE NO ACTION,  
    P_IVA_FINANZIATORE varchar(31),  
    FOREIGN KEY (P_IVA_FINANZIATORE) REFERENCES Finanziatore(P_IVA_FINANZIATORE)  
        ON DELETE CASCADE  
        ON UPDATE NO ACTION,  
    codF INT NOT NULL,  
    FOREIGN KEY (codF) REFERENCES Film(codF)  
        ON DELETE CASCADE  
        ON UPDATE NO ACTION  
);
```

```
CREATE TABLE if not exists Film_Membro_Troupe(  
    codF INT NOT NULL,  
    FOREIGN KEY (codF) REFERENCES Film(codF)  
        ON DELETE CASCADE  
        ON UPDATE NO ACTION,  
    CF varchar(18) NOT NULL,  
    FOREIGN KEY (CF) REFERENCES MembroTroupe(CF)  
        ON DELETE CASCADE  
        ON UPDATE NO ACTION,  
    PRIMARY KEY(codF, CF)  
);
```

```
CREATE TABLE if not exists ScenaCiak(  
    codScena int primary key,  
    noteDiProduzione varchar(255),  
    rullo int NOT NULL,  
    numRiprese int NOT NULL,  
    durataOre float NOT NULL check(durataOre > 0),  
    costoAffittoGiornaliero float(5),  
    codF INT NOT NULL,
```

```
FOREIGN KEY (codF) REFERENCES Film(codF)
ON DELETE CASCADE
ON UPDATE NO ACTION
);

CREATE TABLE if not exists MembroTroupeScena(
    codScena INT NOT NULL,
    FOREIGN KEY (codScena) REFERENCES ScenaCiak(codScena)
    ON DELETE CASCADE
    ON UPDATE NO ACTION ,
    CF varchar(18) NOT NULL,
    FOREIGN KEY (CF) REFERENCES MembroTroupe(CF)
    ON DELETE CASCADE
    ON UPDATE NO ACTION,
    PRIMARY KEY(codScena, CF)
);

CREATE TABLE if not exists Magazzino(
    numMagazzino int Primary Key,
    codInd INT NOT NULL,
    FOREIGN KEY (codInd) REFERENCES Indirizzo(codInd)
    ON DELETE CASCADE
    ON UPDATE NO ACTION
);

CREATE TABLE if not exists PosizioneMagazzino(
    codP int,
    numMagazzino INT NOT NULL,
    scaffale int NOT NULL,
    percorso varchar(1) NOT NULL,
    PRIMARY KEY (codP, numMagazzino),
    FOREIGN KEY (numMagazzino) REFERENCES Magazzino(numMagazzino)
    ON DELETE CASCADE
    ON UPDATE NO ACTION
);

CREATE TABLE if not exists Costume(
    codC int Primary Key,
    tipo varchar(22) NOT NULL CHECK(tipo='epoca' OR tipo='contemporaneo' OR
tipo='fantasia'),
    descrizione varchar(255) NOT NULL,
    CF varchar(18) NOT NULL,
    FOREIGN KEY (CF) REFERENCES MembroTroupe(CF)
    ON DELETE CASCADE
    ON UPDATE NO ACTION,
    codP INT NOT NULL,
    -- FOREIGN KEY (codP) REFERENCES PosizioneMagazzino(codP)
    FOREIGN KEY (codP) REFERENCES PosizioneMagazzino(codP)
    ON DELETE CASCADE
    ON UPDATE NO ACTION
);

CREATE TABLE if not exists StilistaCostume(
    CF varchar(18) NOT NULL,
```



```
FOREIGN KEY (CF) REFERENCES MembroTroupe(CF)
ON DELETE CASCADE
ON UPDATE NO ACTION,
codC INT NOT NULL,
FOREIGN KEY (codC) REFERENCES Costume(codC)
ON DELETE CASCADE
ON UPDATE NO ACTION,
PRIMARY KEY(CF, codC)
);
```

```
CREATE TABLE if not exists CostumeScena(
    codC INT NOT NULL,
    FOREIGN KEY (codC) REFERENCES Costume(codC)
    ON DELETE CASCADE
    ON UPDATE NO ACTION,
    codScena INT NOT NULL,
    FOREIGN KEY (codScena) REFERENCES ScenaCiak(codScena)
    ON DELETE CASCADE
    ON UPDATE NO ACTION,
    PRIMARY KEY(codC, codScena)
);
```

```
CREATE TABLE if not exists OggettiDiScena(
    cod0 int Primary key,
    tipo varchar(21) CHECK (tipo in
('arredo','maschere','armi','mobili','strumentoMusicale','motori')),
    descrizione varchar(255) NOT NULL,
    codP INT NOT NULL,
    FOREIGN KEY (codP) REFERENCES PosizioneMagazzino(codP)
    ON DELETE CASCADE
    ON UPDATE NO ACTION
);
```

```
CREATE TABLE if not exists OggettoScena(
    cod0 INT NOT NULL,
    FOREIGN KEY (cod0) REFERENCES OggettiDiScena(cod0)
    ON DELETE CASCADE
    ON UPDATE NO ACTION,
    codScena INT NOT NULL,
    FOREIGN KEY (codScena) REFERENCES ScenaCiak(codScena)
    ON DELETE CASCADE
    ON UPDATE NO ACTION,
    PRIMARY KEY(cod0, codScena)
);
```

```
CREATE TABLE if not exists Ditta(
    P_IVA_DITTA varchar(31) Primary Key,
    nome varchar(41) NOT NULL,
    codInd INT NOT NULL,
    FOREIGN KEY (codInd) REFERENCES Indirizzo(codInd)
    ON DELETE CASCADE
```

```
        ON UPDATE NO ACTION
    );

CREATE TABLE if not exists Acquisto(
    idAcquisto int Primary Key,
    data date NOT NULL,
    prezzoTotale float(8) NOT NULL,
    P_IVA_DITTA varchar(11) NOT NULL,
    FOREIGN KEY (P_IVA_DITTA) REFERENCES Ditta(P_IVA_DITTA)
    ON DELETE CASCADE
    ON UPDATE NO ACTION
);

CREATE TABLE if not exists AcquistoCostume(
    codC INT NOT NULL,
    FOREIGN KEY (codC) REFERENCES Costume(codC)
    ON DELETE CASCADE
    ON UPDATE NO ACTION,
    idAcquisto INT NOT NULL,
    FOREIGN KEY (idAcquisto) REFERENCES Acquisto(idAcquisto)
    ON DELETE CASCADE
    ON UPDATE NO ACTION,
    prezzo float(8) NOT NULL,
    PRIMARY KEY(codC, idAcquisto)
);

CREATE TABLE if not exists AcquistoOggetto(
    codO INT NOT NULL,
    FOREIGN KEY (codO) REFERENCES OggettiDiScena(codO)
    ON DELETE CASCADE
    ON UPDATE NO ACTION,
    idAcquisto INT NOT NULL,
    FOREIGN KEY (idAcquisto) REFERENCES Acquisto(idAcquisto)
    ON DELETE CASCADE
    ON UPDATE NO ACTION,
    prezzo float(8) NOT NULL,
    PRIMARY KEY(codO, idAcquisto)
);
```

## 7 Interfaccia grafica

---

### 7.1 Descrizione dell'architettura utilizzata

---

Si sviluppa una applicazione molto semplice per la gestione del database in linguaggio Java, che renda possibile la messa in pratica delle operazioni richieste dalle varie viste. L'approccio verso il DB è gestito tramite JDBC. Il DB risiede in locale e usa MySQL come DBMS. L'applicativo consentirà all'admin di poter eseguire tutte le operazioni necessarie per l'inserimento in database di un film che andrà a comporre quello che sarà il catalogo e visualizzare tutte le informazioni a esso relative.

## 7.2 Interfaccia di inserimento

L'applicativo si avvierà con quella che è la schermata d'inserimento dei dati, che permetterà all'amministratore l'inserimento di tutti i dati riguardante un film, oltre i film stessi. In questa schermata sarà possibile inserire:

- Film
- Enti
- Finanziatori
- Incassi
- Indirizzi
- MembriTroupe
- Sponsor
- Schermata inserimento di un film

The screenshot shows a web application window titled "Database Cinematografico". It features a navigation bar with tabs for "Insert", "visualization", and "query". Below this, there are buttons for "Film", "Indirizzo", "Enti", "Operatore", "Sponsor", "Finanziatore", and "Incasso". The "Film" button is selected.

The main content area is divided into two panels. The left panel, titled "FILM", contains a form for inserting film data with the following fields:

- Ttolo
- Genere
- Durata
- Data Uscita
- ID serie letteraria

Below the form is an "Insert" button.

The right panel displays a table with the following columns: "codF", "titolo", "genere", "durata", and "dataUscita". The table contains four rows of data:

codF	titolo	genere	durata	dataUscita
1	Star Wars: Episodio VI - Il ritorno dello Jedi	fantascienza	134	1983-05-25
2	Il signore degli anelli, la compagnia dell anello	fantascienza	134	1983-05-25
3	La Mummia	fantascienza	120	1999-05-27
4	ritorno al futuro II	fantascienza	108	1989-10-05

At the bottom right of the application window is a "Refresh" button.

- Schermata inserimento di un ente

Database Cinematografico

Insert

visualization

query

Film

Indirizzo

Enti

Operatore

Sponsor

Finanziatore

Incasso

ENTI

P.IVA

Nome

Codice Indirizzo

Insert

P_IVA	nome	codInd
40365320379	UCI Milano	18302

Refresh

- Schermata inserimento di un finanziatore

Database Cinematografico

Insert

visualization

query

Film

Indirizzo

Enti

Operatore

Sponsor

Finanziatore

Incasso

FINANZIATORE

Seleziona Film

P.IVA

Nome

Codice Indirizzo

Percentuale Guadagno

Insert

Nessuna colonna nella tabella

Refresh

- Schermata inserimento di un incasso

Database Cinematografico

Insert

visualization

query

Film

Indirizzo

Enti

Operatore

Sponsor

Finanziatore

Incasso

INCASSO SETTIMANALE

Seleziona Film

Data Inizio

Data Fine

Incasso

Cod. Indirizzo sede t.

Insert

Nessuna colonna nella tabella

Refresh

- Schermata inserimento di un indirizzo

Database Cinematografico

Insert

visualization

query

Film

Indirizzo

Enti

Operatore

Sponsor

Finanziatore

Incasso

INDIRIZZO

Codice Indirizzo

Città

Via

Civico

CAP

Insert

codInd	città	via	civico	CAP
10185	Sesto Federica nell'e	Rotonda Veneziano	50	90813
12341	Fagotto lido	Canale Giampiero	36	89071
12346	Lake Nyahmouth	McGlynn Brooks	418	70444
13611	Sobrero calabro	Strada Pennetta	2	15942
13859	New York	Adams Street	117	11201
14261	Piacentini sardo	Borgo Versace	9	98829
15293	Vermont	Jayden Villages	3444	54808
16378	Caverzere	Armando Diaz	1	30014
16395	savignano	Piazza Metropolis	18	47039
16957	New York	Empire State	23	34534
17349	Arielleport	Ashton Tunnel	645	48937
17386	Borgo Gionata	Contrada Siffredi	260	76300
17439	Maine	McDermott Manors	339	33812
18302	Milano	Corso Semprione	39	20145
18692	Felipeport	Nicolas Branch Suite 189	78261	97077
19196	San Ramona del friuli	Incrocio Faranda	9	67111
19243	South Vesperport	Alfonso del...	650	20770

Refresh

- Schermata inserimento di un operatore

The screenshot shows a web application titled "Database Cinematografico". At the top, there are three tabs: "Insert", "visualization", and "query". Below these, a horizontal menu contains buttons for "Film", "Indirizzo", "Enti", "Operatore", "Sponsor", "Finanziatore", and "Incasso". The "Operatore" button is currently selected. The main content area is split into two panels. The left panel, titled "Operatore Troupe", contains a form with the following fields: a dropdown menu labeled "Seleziona Film", text input fields for "CF", "Nome", "Cognome", "IBAN", "Data Nascita" (with a calendar icon), "Telefono", "Codice Indirizzo", and "Percentuale Contributo". An "Insert" button is at the bottom of this form. The right panel is a large white area displaying the message "Nessuna colonna nella tabella". At the bottom right of the application window, there is a "Refresh" button.

- Schermata inserimento di uno sponsor

The screenshot shows the same "Database Cinematografico" application, but with the "Sponsor" button selected in the horizontal menu. The left panel, titled "SPONSOR", contains a form with the following fields: a dropdown menu labeled "Seleziona Film", text input fields for "P.IVA", "Nome", and an "Insert" button. The right panel remains the same, displaying "Nessuna colonna nella tabella". The "Refresh" button is still present at the bottom right.

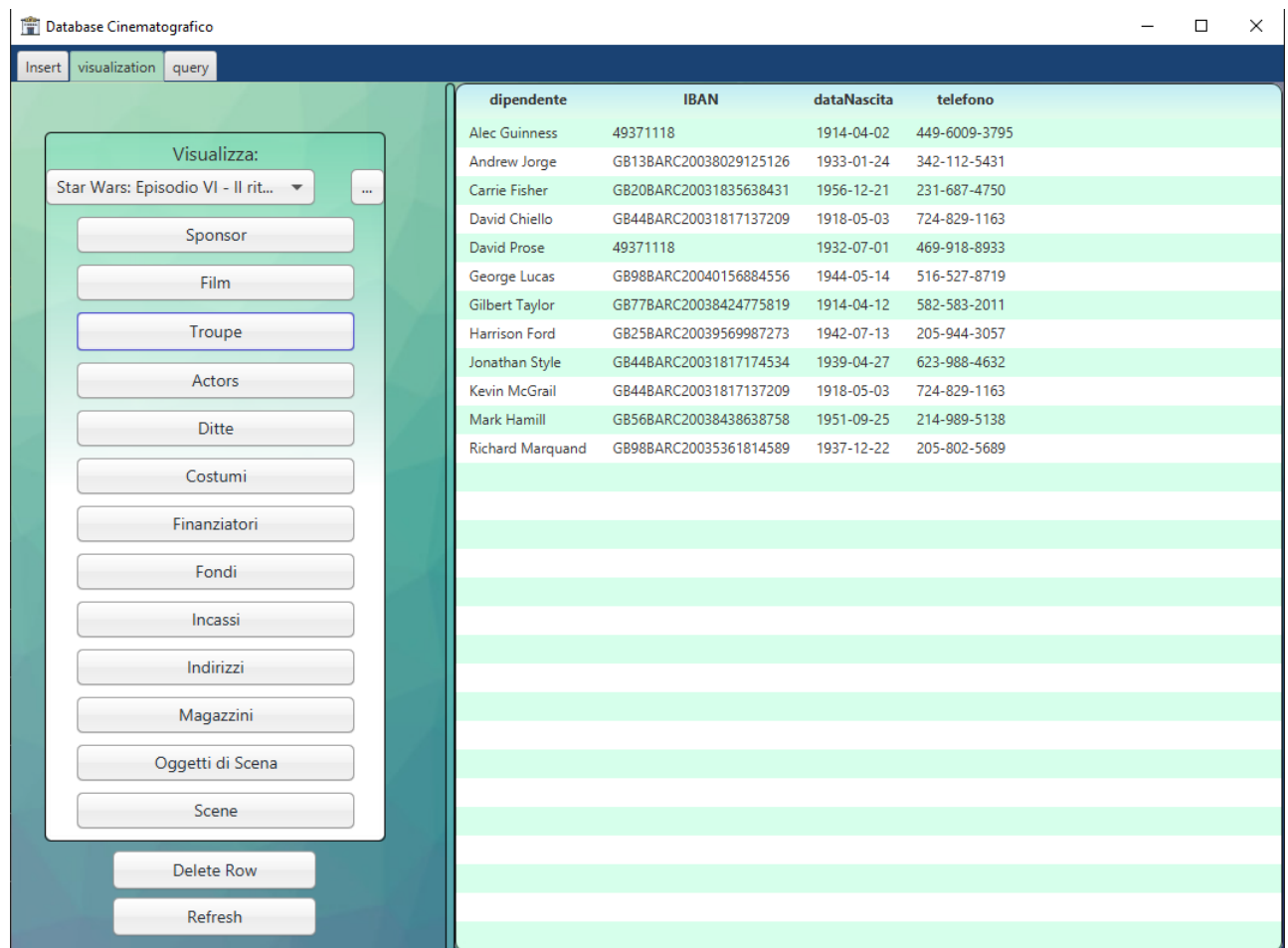
## 7.3 Interfaccia di visualizzazione

---

In questa interfaccia sarà possibile per l'amministratore visualizzare tutti i dati contenuti nel database per un determinato film. Come esempio prenderemo la visualizzazione dei dati riguardanti il film Star wars. Qui l'utente sarà in grado di visualizzare i dati relativi a:

- Film
- Enti
- Finanziatori
- Incassi
- Indirizzi
- MembriTroupe
- Sponsor
- Attori
- Ditte
- Costumi
- Oggetti
- Magazzini
- Scene

- Schermata di visualizzazione



## 7.4 Interfaccia per operazioni specifiche

In questa interfaccia l'amministratore potrà effettuare tutte quelle operazioni specifiche di visualizzazione di dati particolare.

Qui l'utente sarà in grado di visualizzare:



- [illegible]

- Database Cinematografico

Insert

visualization

query

Query Speciali

Star Wars... ▾

OK

Ricavi

Refresh

Stipendio Membri Trou...

Profitto Finanziatori

Scene in film ▾

Dipendenti Scena

Magazzino ▾

Profitto Produttori

Incasso Film

Costumi Scena

Oggetti Scena

Oggetti in Magazzino

TotaleIncasso

163874957

- [illegible]

- Insert

visualization

query

Star Wars... ▾

OK

Ricavi

Refresh

Stipendio Membri Trou...

Profitto Finanziatori

Scene in film ▾

Dipendenti Scena

Magazzino ▾

Profitto Produttori

Incasso Film

Costumi Scena

Oggetti Scena

Oggetti in Magazzino

nome	cognome	percentualeContributo	guadagno	nomeRuolo
George	Lucas	1	1638749.57	Sceneggiatore
George	Lucas	1	1638749.57	Produttore Esecutivo
George	Lucas	1	1638749.57	Produttore
Gilbert	Taylor	0.8	1310999.6755354174	Aiuto Regista
Richard	Marquand	0.47	770212.2959464582	Regista

- Ricavi di un film

Database Cinematografico

Insert

visualization

query

Query Speciali

Il signore ...

OK

Ricavi

Refresh

Stipendio Membri Trou...

Profitto Finanziatori

Scene in film

Dipendenti Scena

Magazzino

Profitto Produttori

Incasso Film

Costumi Scena

Oggetti Scena

Oggetti in Magazzino

Ricavo

40253111.36803854

- Dipendenti che hanno partecipato scena di un film

[illegible]

- [illegible]

- [illegible]

- Oggetti presenti in magazzino

Database Cinematografico

Insertvisualizationquery

Query Speciali

Star Wars...OKRicaviRefresh

Stipendio Membri Trou...Profitto FinanziatoriScene in filmDipendenti ScenaMagazzino

Profitto ProduttoriIncasso FilmCostumi ScenaOggetti ScenaOggetti in Magazzino

tipo	descrizione	scaffale	percorso
armi	Cannone blaster rotante Z-6	2	S
armi	Fucile Blaster E-5	2	S
armi	Spada rossa Darth vader	2	P
armi	Spada blu Di anakin	11	B
arredo	oggetto di scena n.13	7	Y
maschere	oggetto di scena n.15	2	F
armi	oggetto di scena n.5	2	A
maschere	oggetto di scena n.7	3	Z
mobili	oggetto di scena n.10	6	X
strumentoMusicale	oggetto di scena n.1	1	A
strumentoMusicale	oggetto di scena n.4	4	Q
armi	oggetto di scena n.15	6	Q
arredo	oggetto di scena n.18	9	G
strumentoMusicale	oggetto di scena n.13	7	S
arredo	oggetto di scena n.3	2	I
motori	oggetto di scena n.12	1	C
maschere	oggetto di scena n.11	5	W
motori	oggetto di scena n.2	6	M
strumentoMusicale	oggetto di scena n.17	3	W
mobili	oggetto di scena n.7	5	S
mobili	oggetto di scena n.8	9	N
arredo	oggetto di scena n.1	6	O

# Special thanks

libreria python per generare tutto

generatore di indirizzi random

generatore di IBAN random

generatore di numeri telefono random

generatore di P.IVA random

indirizzo e persona