



## **UNIVERSITÀ DEGLI STUDI DI UDINE**

DIPARTIMENTO DI SCIENZE MATEMATICHE, INFORMATICHE E FISICHE CORSO DI  
LAUREA TRIENNALE IN INFORMATICA

### **PROGETTAZIONE E IMPLEMENTAZIONE DI UN DATABASE PER UN ESERCIZIO COMMERCIALE**

Docenti:

Chiar.mo Prof. Luca Geatti

Chiar.mo Prof. Angelo Montanari

Partecipanti al progetto:

Ioana Larisa DINULESCU

Matricola n. 157046

Federico DEL BELLO

Matricola n. 152732

Luca SCAINI

Matricola n. 159487

ANNO ACCADEMICO 2023/2024

<b>BASE DI DATI DI UN ESECIZIO COMMERCIALE– GRUPPO 15</b>	<b>2</b>
<b>1 Analisi dei requisiti</b>	<b>2</b>
1.1 Testo	2
1.2 Analisi del testo	2
<b>2 PROGETTAZIONE CONCETTUALE</b>	<b>2</b>
2.1 Schema Entità-Relazioni	2
2.2 Vincoli di integrità	3
2.3 Analisi dei cicli	3
2.4 Tabella dei volumi	4
2.5 Tabella delle frequenze	5
2.6 Analisi delle ridondanze	5
2.7 Conclusioni	6
<b>3 PROGETTAZIONE LOGICA</b>	<b>7</b>
3.1 Notazioni	7
3.2 Traduzione	7
<b>4 PROGETTAZIONE FISICA: TRADUZIONE IN SQL</b>	<b>10</b>
4.1 Creazione del database	10
4.2 Vincoli di dominio	10
4.3 Query	10
4.4 Trigger	11
<b>5 ANALISI DEI DATI CON LINGUAGGIO R</b>	<b>13</b>
5.1 Andamento dei noleggi per l'anno 2021	13
5.2 Analisi dei noleggi per casa di produzione	13
5.3 Genere degli attori protagonisti	14

# BASE DI DATI DI UN ESERCIZIO COMMERCIALE PER NOLEGGI DI DVD

## 1 Analisi dei requisiti

### 1.1 Testo

Si voglia modellare il seguente insieme di informazioni riguardanti un esercizio commerciale che si occupa del noleggio di dvd di film.

- Ogni **film** sia caratterizzato da un *codice identificativo*, dal titolo, da una o più case di produzione (si assuma che un film possa essere co-prodotto da più case di produzione), dall'anno di uscita, da uno o più generi (comico, poliziesco, romantico, ...), dal/i regista/i, da un insieme di attori principali (alcuni film, come i film di animazione, possono essere privi di attori) e da una breve descrizione testuale.
- Ogni **casa di produzione** sia caratterizzata da un nome, che la identifica univocamente, da un amministratore delegato, dall'anno di fondazione e da un recapito postale.
- Di ogni film siano disponibili uno o più **dvd**, ciascuno identificato da un *numero progressivo*. Per ogni dvd, si vuole tener traccia di tutti i noleggi. Ogni **noleggio** sia caratterizzato dal cliente che ha noleggiato il dvd e dall'intervallo di tempo (giorno iniziale e giorno finale) del noleggio. Si assuma che ogni dvd possa essere noleggiato per un minimo di 2 giorni e un massimo di 7. Si assuma anche che uno stesso cliente possa noleggiare più volte uno stesso dvd.
- Ogni **cliente** sia identificato dal suo *codice fiscale* e sia caratterizzato dal nome, dal cognome, dalla data di nascita, da un recapito telefonico, da un indirizzo e-mail (opzionale) e da un recapito postale.

Si definisca uno schema Entità-Relazioni che descriva il contenuto informativo del sistema, illustrando con chiarezza le eventuali assunzioni fatte. Lo schema dovrà essere completato con attributi ragionevoli per ciascuna entità (identificando le possibili chiavi) e relazione. Vanno specificati accuratamente i vincoli di cardinalità e partecipazione di ciascuna relazione. Si definiscano anche eventuali regole di derivazione e/o vincoli di integrità necessari per codificare alcuni dei requisiti attesi del sistema.

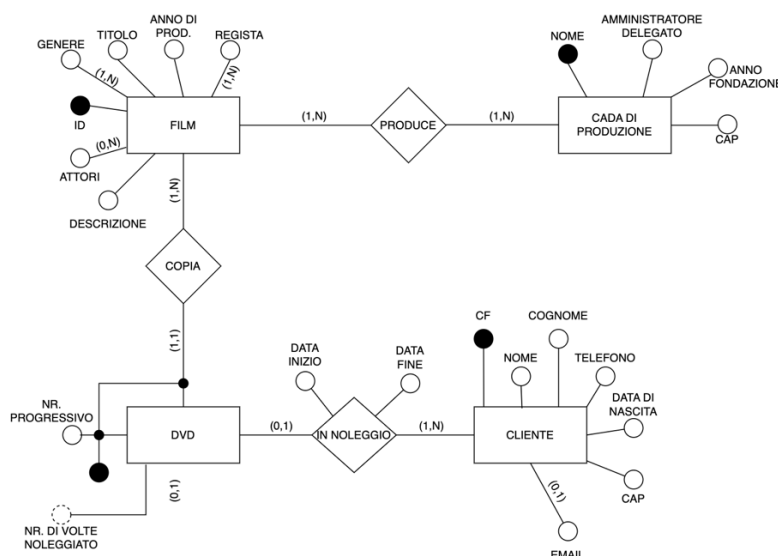
### 1.2 Analisi del testo

Dopo una prima analisi sono state evidenziate in grassetto le **entità**, in corsivo i campi *chiave primaria* e con una sottolineatura gli attributi.

## 2 PROGETTAZIONE CONCETTUALE

### 2.1 Schema Entità-Relazioni

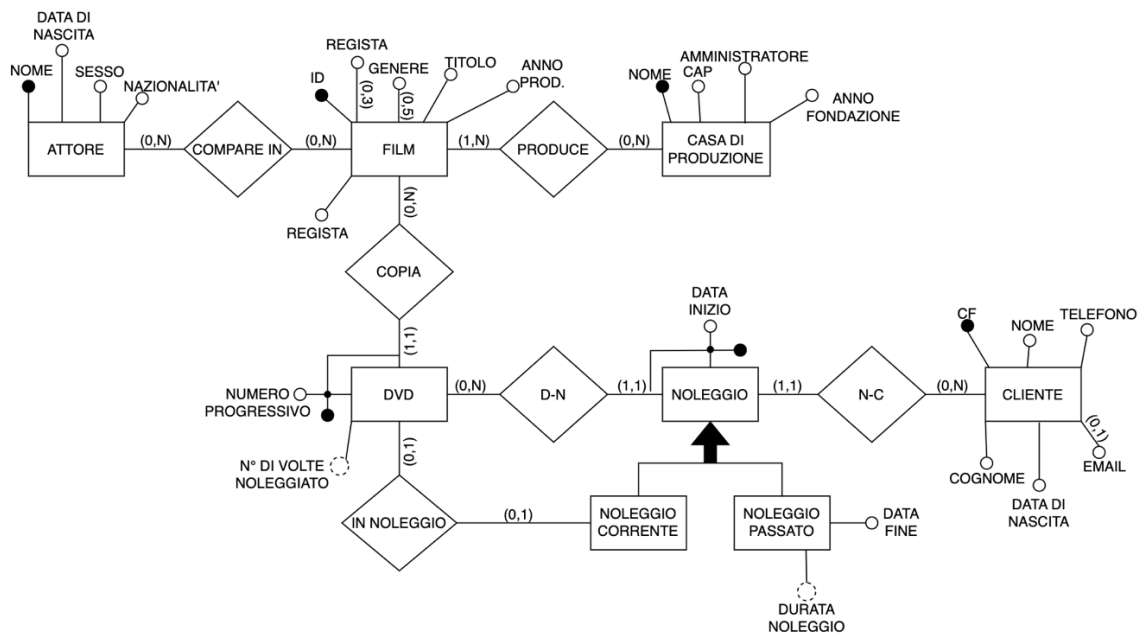
Per la realizzazione dello schema E-R è stata utilizzata inizialmente la strategia inside-out, ottenendo in questo modo il seguente risultato:



Questo modello non rispettava però pienamente tutti i requisiti dell'esercizio, perché non si riusciva a creare uno storico, con le rispettive date di inizio e fine noleggio.

Quindi usando la strategia top-down, per creare lo storico, si è deciso di reificare la relazione “noleggio” in entità, con i rispettivi figli “inizio noleggio” e “fine noleggio”.

Inoltre, l'attributo “attore” dell'entità “film” è stato trasformato in entità in quanto essendo originariamente un attributo a valori multipli, avrebbe potuto creare problemi nella trasposizione in SQL.



Come si può notare dallo schema, la maggior parte delle entità ha una sola chiave primaria, ma ci sono due eccezioni:

Nel caso di “DVD” è stata inserita una chiave esterna che vede come entità forte “FILM” e come entità debole “DVD”, poiché il DVD ha un numero progressivo che lo identifica fra le copie di un unico film, per identificare quindi i DVD in maniera univoca, si devono considerare sia il nr. progressivo che il film.

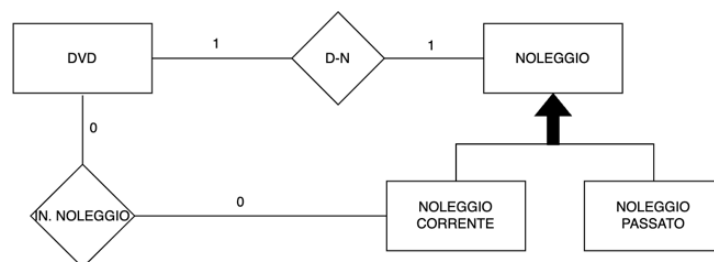
Un'altra chiave esterna vede come entità forte “DVD” e come entità debole “NOLEGGIO” perché il noleggio è identificato della data del suo inizio ma anche dal dvd noleggiato. Inizialmente è stato preso in considerazione di mettere come entità forte “CLIENTE”, ma in quel caso un cliente avrebbe potuto noleggiare un solo DVD alla volta.

## 2.2 Vincoli di integrità

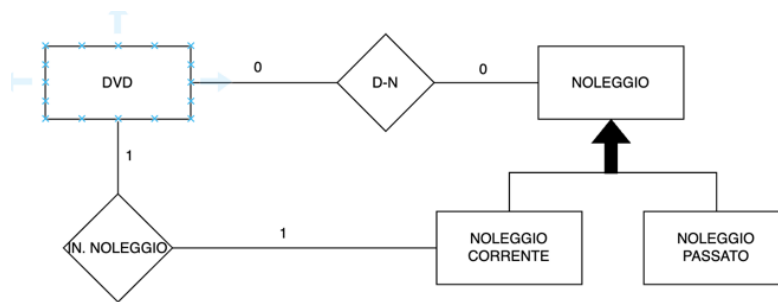
- Due edizioni di uno stesso film non possono essere state prodotte nello stesso anno
- La durata del noleggio deve essere di minimo 2 giorni e massimo di 7. Quindi  $2 < \text{durata noleggio} < 7$
- Se un DVD è presente in “NOLEGGIO CORRENTE” deve necessariamente essere presente anche in “NOLEGGIO” (vedi analisi del ciclo).

## 2.3 Analisi dei cicli

È presente un ciclo, quindi sono state analizzate le due situazioni in cui si sarebbero potuti verificare problemi. Prima situazione:



In questo caso sarebbe normale perché un DVD può essere stato noleggiato in passato, e quindi essere presente nello storico, ma non essere più in NOLEGGIO CORRENTE perché è già stato restituito



Questa situazione invece non è accettabile perché un DVD attualmente in noleggio, deve essere necessariamente presente anche in “NOLEGGIO”. Per risolvere questo problema c’è un vincolo di integrità che impedisce di inserire un nuovo noleggio solo in “NOLEGGIO CORRENTE” e non anche in “NOLEGGIO”.

## 2.4 Tabella dei volumi

Nella creazione di un database è necessario tenere sempre conto dell’ambito applicativo e della quantità di dati che la struttura andrà a gestire, questo per poter avere un’idea di come organizzare al meglio la struttura, le relazioni e la gestione dei dati nel database.

Il primo passo è stato organizzare la tabella dei volumi, in cui sono stati inseriti dei dati approssimativi scelti in base a una ricerca e a procedimenti logici, che verranno spiegati in seguito, successivamente è stata stimata la tabella delle frequenze, la quale va ad indicare la frequenza stimata con cui verranno eseguite le operazioni nel database.

Concept	Type	Volume
Case produttrici	E	6
Film	E	100
DVD	E	1000
Noleggio	E	100 000
Noleggio passato	E	100 000
Noleggio corrente	E	1000
cliente	E	15 000
Produce	R	600
Copia	R	1000
D-N	R	100 000
In noleggio	R	1000
N-C	R	100 000

Nella tabella vengono riportati i concept, ovvero gli elementi che vanno a comporre lo schema relazionale, viene di seguito specificato il loro ruolo nello schema, quindi se sono entità o relazioni.

Come ultimo passaggio sono stati stimati i volumi, ovvero la mole di dati che si prevede il database dovrà gestire. Per la stima dei volumi sono stati scelti numeri ponderati in base a ricerche online.

Il numero di case produttrici è sei, nonostante esistano molte più case produttrici famose, ma considerando che la maggioranza delle società produttrici di film appartengono a grandi conglomerati di società denominate con lo stesso appellativo, è stato deciso di fare riferimento direttamente a quelli.

Di questi sei conglomerati sono stati stimati venti film a conglomerato e dieci copie di ogni film, ottenendo in questo modo 1000 dvd totali. Per l’entità “noleggio” sono stati stimati 100 000 noleggi totali e ovviamente altrettanti noleggi passati nello storico. Sono stati poi stimati 1000 noleggi nel caso improbabile in cui tutte e 1000 le copie del negozio vengano noleggiate allo stesso momento.

Sono stati assunti una totalità di 15 000 clienti, numero attendibile se si considera che circa il 15% della popolazione della città noleggi un film (è stata presa in considerazione la popolazione di Udine).

Per quanto riguarda le relazioni, sono stati stimati i volumi in base ai vincoli di cardinalità dello schema relazionale; infatti, su “copia” è stato messo 1000 perché in relazione (1-1) con l’entità “DVD”, allo stesso modo “D-N” è in relazione con “noleggio” mentre la relazione “in noleggio” è legata all’entità “noleggio corrente”.

## 2.5 Tabella delle frequenze

Descrizione operazioni	Operazione	tipo	Frequenza
Inserimento di un nuovo noleggio	OP1	I	50 volte al giorno
Tutti i cf delle persone che hanno noleggiato un film di Spielberg	OP2	I	1 volta a settimana
Tutte le case di produzione che hanno prodotto almeno 3 film	OP3	I	1 volta al mese
Eliminazione delle copie che sono state noleggate più di 10 000 volte	OP4	P	1 volta al mese
Aggiornamento del numero di telefono di un cliente	OP5	I	2 volte all'anno
Media delle durate dei noleggi conclusi	OP6	P	1 volta al mese
Calcolo del best seller della settimana, film più noleggiato	OP7	P	1 volta a settimana
Numero noleggi per copia	OP8	P	1 volta al mese
Restituzione di un film noleggiato	OP9	I	50 volte al giorno

Dove I sta per operazione interattiva e P per operazione programmata.

Durante la stesura della tabella è stato assegnato a ogni operazione un identificativo "OP N" per rendere più snello e immediato lo sviluppo dell'analisi delle ridondanze che verrà eseguito di seguito.

Sono state scelte queste operazioni per poter integrare tutte le tipologie possibili di interazioni che verranno eseguite nel database, ovvero cancellazione, inserimento, aggiornamento e interrogazione, con annesse frequenze di utilizzo.

## 2.6 Analisi delle ridondanze

Un attributo/entità ridondante è un attributo derivabile da altri.

Metodo generale di risoluzione:

Per determinare se mantenere un attributo ridondante può essere utile fare un'analisi dei costi delle operazioni che coinvolgono tale attributo nel seguente formato:

$\text{CostoOperazione} = (\text{TipologiaOperazione} * \text{FrequenzaOperazione}) * \text{TupleCoinvolte}$

Dove la tipologia Operazione è espressa dalla tabella sottostante:

Tipologia Operazione	Costo ( $\mu$ )
<b>Lettura (L)</b>	1
<b>Scrittura (S)</b>	2

Il costo dell'operazione viene valutato in presenza ed in assenza dell'attributo ridondante per poi confrontare i risultati e determinare se la presenza o meno di tale attributo è conveniente a livello di peso delle operazioni eseguite.

### Calcoli

In questo caso sono presenti i seguenti attributi ridondanti:

1. NumeroDiNoleggi  
Attributo di DVD che può essere ricavato dal numero di volte che lo specifico DVD appare nell'entità noleggio.

Costo OP1 + Costo OP8

- Valutazione con ridondanza:  
 $OP1 = \text{InserimentoNuovaTupla}\{\text{Entità coinvolta: Noleggio; Tipo: S; Tuple coinvolte: 1}\} + \text{AggiornamentoDatoRidondante}\{\text{Entità coinvolta: DVD; Tipo: S; Tuple coinvolte: 1}\} * \text{FrequenzaOP1}$   
 $OP1 = ((2*1)+(2*1))*50 = 200 \mu/\text{giorno} = 6'000 \mu/\text{mese}$   
 $OP8 = \text{ConteggioNoleggiCopia}\{\text{Entità coinvolta: DVD; Tipo: L; Tuple coinvolte: 1'000}\} * \text{FrequenzaOP8}$   
 $OP8 = (1*1)*1'000 = 1000 \mu/\text{mese}$   
 $\text{Totale} = OP1 + OP8 = 6'000 + 1'000 = 7'000 \mu/\text{mese}$
- Valutazione senza ridondanza:  
 $OP1 = \text{InserimentoNuovaTupla}\{\text{Entità coinvolta: Noleggio; Tipo: S; Tuple coinvolte: 1}\} * \text{FrequenzaOP1}$   
 $OP1 = (2*1)*50 = 100 \mu/\text{giorno} = 3'000 \mu/\text{mese}$   
 $OP8 = \text{ConteggioNoleggiCopia}\{\text{Entità coinvolta: Noleggio; Tipo: L; Tuple coinvolte: 100'000}\} * \text{FrequenzaOP8}$   
 $OP8 = (1*100'000)*1 = 100'000 \mu/\text{mese}$   
 $\text{Totale} = OP1 + OP8 = 3'000 + 100'000 = 103'000 \mu/\text{mese}$

## 2. DurataNoleggio

Attributo dell'Entità noleggio che può essere ricavato da DataFine - DataInizio  
 Costo OP9 + Costo OP6

- Valutazione con ridondanza:  
 $OP9 = \text{Aggiornamento}\{\text{Entità coinvolta: Noleggio; Tipo: S; Tuple coinvolte: 1}\} * \text{FrequenzaOP9}$   
 $OP9 = (2*1)*50 = 100 \mu/\text{giorno} = 3'000 \mu/\text{mese}$   
 $OP6 = \text{LetturaDurataNoleggio}\{\text{Entità coinvolta: Noleggio; Tipo: L; Tuple coinvolte: 1'500}\} * \text{FrequenzaOP6}$   
 $OP6 = (1*1'500)*1 = 1'500 \mu/\text{mese}$   
 $\text{Totale} = OP9 + OP6 = 3'000 + 1'500 = 4'500 \mu/\text{mese}$
- Valutazione senza ridondanza:  
 $OP9 = \text{Aggiornamento}\{\text{Entità coinvolta: Noleggio; Tipo: S; Tuple coinvolte: 1}\} * \text{FrequenzaOP9}$   
 $OP9 = (2*1)*50 = 100 \mu/\text{giorno} = 3'000 \mu/\text{mese}$   
 $OP6 = \text{LetturaDataInizio}\{\text{Entità coinvolta: Noleggio; Tipo: L; Tuple coinvolte: 1'500}\} + \text{LetturaDataFine}\{\text{Entità coinvolta: Noleggio; Tipo: L; Tuple coinvolte: 1'500}\} * \text{FrequenzaOP6}$   
 $OP6 = (1*1)*1'500 = 1'500 \mu/\text{mese}$   
 $\text{Totale} = OP9 + OP6 = 3'000 + 1'500 = 4'500 \mu/\text{mese}$


## 2.7 Conclusioni

Dopo l'analisi eseguita precedentemente si può concludere che è conveniente mantenere l'elemento ridondante NumeroDiNoleggi dell'entità DVD, mentre per l'attributo DurataNoleggio dell'entità Noleggio risulta indifferente se tenerlo o meno, pertanto abbiamo deciso di toglierlo.

### 3 PROGETTAZIONE LOGICA

#### 3.1 Notazioni

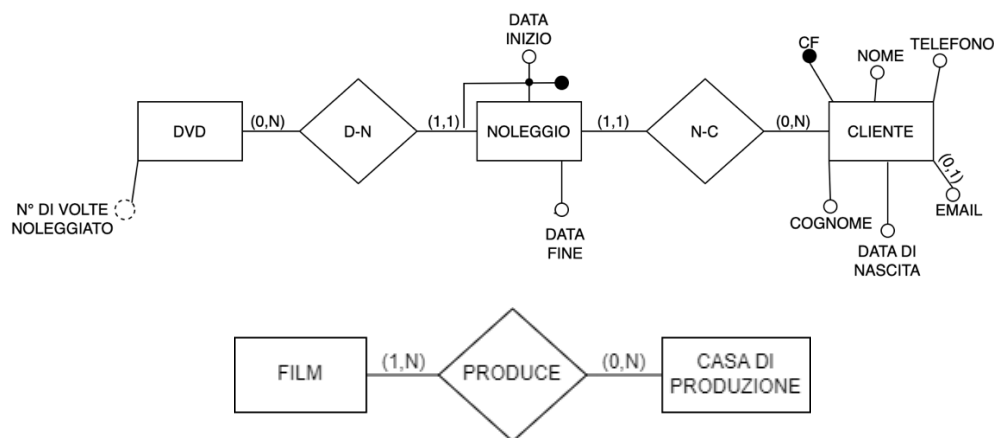
Per rendere più chiara la notazione è stato deciso di esprimere gli attributi ed i loro vincoli sottoforma di tabella dove:

- Sulle righe ci sono gli attributi dell'entità descritta;
- Sulle colonne ci sono i vincoli che devono rispettare i rispettivi attributi (✓);
- La notazione  indica che i valori devono rispettare il vincolo come insieme.



Le chiavi primarie sono indicate con la sottolineatura.

#### 3.2 Traduzione

Per la traduzione del diagramma relazionale è stato modificato lo schema E-R reificando la specializzazione di "noleggio", lasciando solamente il padre.



Film{ id, titolo, annoDiProduzione, generi, regista, descrizione }

Attributo di Film	Primary Key	Forein Key	FK riferimento	NOTNULL	UNIQUE
<b>Id</b>	✓	✗		✓	✓
<b>Titolo</b>	✗	✗		✓	✓ 
<b>annoDiProduzione</b>	✗	✗		✓	✓ 
<b>Generi</b>	✗	✗		✓	✗
<b>Regista</b>	✗	✗		✓	✗
<b>Descrizione</b>	✗	✗		✓	✗

CasaDiProduzione{ nome, amministratore, annoDiFondazione, cap }

Attributo di CasaDiProduzione	Primary Key	Forein Key	FK riferimento	NOTNULL	UNIQUE
<b>nome</b>	✓	✗		✓	✓
<b>amministratore</b>	✗	✗		✓	✗
<b>annoDiFondazione</b>	✗	✗		✓	✗
<b>cap</b>	✗	✗		✓	✗

Produce{ CasaDiProduzione.nome, Film.id }



Attributo di Produce	Primary Key	Foreign Key	FK riferimento	NOTNULL	UNIQUE
<b>NomeCasaDiProduzione</b>			CasaDiProduzione		
<b>IdFilm</b>			Film		



Attore { Nome, DataDiNascita, Sesso, Nazionalità }

Attributo di Attore	Primary Key	Foreign Key	FK riferimento	NOTNULL	UNIQUE
<b>Nome</b>					
<b>DataDiNascita</b>					
<b>Sesso</b>					
<b>Nazionalità</b>					

CompareIn { NomeAttore, IdFilm }

Attributo di CompareIn	Primary Key	Foreign Key	FK riferimento	NOTNULL	UNIQUE
<b>NomeAttore</b>			Attore		
<b>IdFilm</b>			Film		



DVD { progressivo, Film.id, numeroDiNoleggi }

Attributo di DVD	Primary Key	Foreign Key	FK riferimento	NOTNULL	UNIQUE
<b>Progressivo</b>					
<b>IdFilm</b>			Film		
<b>NumeroDiNoleggi</b>					



Noleggio{ dataInizio, progressivoDVD, FilmID, dataFine, ClienteCF, DurataNoleggio }

Attributo di Noleggio	Primary Key	Forein Key	FK riferimento	NOTNULL	UNIQUE
<b>dataInizio</b>					
<b>ProgressivoDVD</b>			DVD		
<b>FilmID</b>			DVD -> Film		
<b>ClienteCF</b>			Cliente		
<b>dataFine</b>					
<b>DurataNoleggio</b>					

Cliente{ cf, nome, cognome, telefono, dataDiNascita, cap, e-mail }

Attributo di Cliente	Primary Key	Forein Key	FK riferimento	NOTNULL	UNIQUE
<b>CF</b>					
<b>Nome</b>					
<b>Cognome</b>					
<b>Telefono</b>					
<b>dataDiNascita</b>					
<b>Cap</b>					
<b>e-mail</b>					

## 4 PROGETTAZIONE FISICA: TRADUZIONE IN SQL

### 4.1 Creazione del database

Per creare e gestire una base di dati, è essenziale avere un server configurato e in esecuzione. Per questo progetto è stato utilizzato il DBMS PostgreSQL.

```
CREATE DATABASE lab_basi;
```

Una volta creato il database lo abbiamo suddiviso logicamente creando lo schema progetto15

```
CREATE SCHEMA progetto15;
```

### 4.2 Vincoli di dominio

ANNO: l'anno di produzione di un film e l'anno di fondazione di una casa di produzione deve essere un numero maggiore di 1850.

```
CREATE DOMAIN dom_anno AS numeric(4,0) CONSTRAINT anno_min CHECK( value > 1850 );
```

ID: i numeri identificativi dei film (Id) e DVD (progressivo) devono essere naturali positivi.

```
CREATE DOMAIN dom_positivo AS numeric(6,0) CONSTRAINT maggiore_di_zero CHECK( value > 0 );
```

GENERI: i generi dei film devono essere compresi nel seguente insieme: {animazione, avventura, biografico, commedia, comico, drammatico, romantico, sci-fi & fantasy, giallo, horror, storico, western}.

```
CREATE TYPE dom_generi AS ENUM ('Animazione', 'Avventura', 'Biografico',  
                                'Commedia', 'Comico', 'Drammatico', 'Romantico', 'Sci-Fi & Fantasy',  
                                'Giallo', 'Horror', 'Storico', 'Western'); /*12 generi*/
```

Una volta creato il database e stabilito i vincoli di dominio sono state definite le tabelle, che sono l'implementazione fisica dello schema relazionale con l'aggiunta del tipo di dato che gli attributi saranno in grado di accettare durante l'utilizzo del database (a seguire un esempio):

```
CREATE TABLE film(  
    id dom_positivo PRIMARY KEY,  
    titolo varchar(255) NOT NULL,  
    anno_di_produzione dom_anno NOT NULL,  
    generi dom_generi NOT NULL,  
    regista varchar(75) NOT NULL,  
    descrizione varchar(1000) NOT NULL,  
    /*vincoli di tabella*/  
    CONSTRAINT film_unico UNIQUE( titolo, anno_di_produzione )  
);
```

Nel file allegato "crea\_db.sql" si possono visionare tutte le operazioni per la creazione del database.

### 4.3 Query

#### prima query:

Seleziona tutti i clienti, identificati univocamente dal loro codice fiscale, che hanno noleggiato almeno una volta un film di Spielberg.

```
SELECT DISTINCT cf FROM cliente AS c1, noleggio AS n1, film AS f1  
    WHERE c1.cf=n1.cliente_cf AND n1.film_id=f1.id AND f1.regista LIKE '%Steven Spielberg%'  
    AND n1.data_fine IS NOT NULL;
```

#### seconda query:

Seleziona tutte le case di produzione che hanno prodotto almeno 3 film. Nel caso specifico di questo database, avendo poche case di produzione e tanti film, quasi tutte le case di produzione vengono selezionate.

```
SELECT DISTINCT nome FROM casa_di_produzione AS cdp, produce AS p1, produce AS p2, produce AS p3
WHERE cdp.nome = p1.nome_casa_di_produzione AND cdp.nome = p2.nome_casa_di_produzione
AND cdp.nome = p3.nome_casa_di_produzione AND p1.id_film <> p2.id_film
AND p2.id_film <> p3.id_film AND p1.id_film <> p3.id_film;
```

#### **terza query:**

Seleziona tutti i clienti, mostrando il loro codice fiscale, che hanno noleggiato un film in cui compare l'attore John Travolta.

```
SELECT cf FROM cliente AS c1, noleggio AS n1, compare_in AS cil, film AS f1
WHERE c1.cf=n1.cliente_cf AND n1.film_id=f1.id AND cil.id_film=f1.id AND cil.nome_attore='John Travolta';
```

#### **quarta query:**

Conta il numero di DVD che sono copie di un film prodotto da una casa di produzione che ha prodotto solamente film western.

```
SELECT id, COUNT(progressivo) FROM dvd, produce AS p1, casa_di_produzione AS cdp1, film AS f1
WHERE dvd.id_film = p1.id_film and cdp1.nome=p1.nome_casa_di_produzione AND f1.id = p1.id_film
AND f1.generi='Western' AND NOT EXISTS(
SELECT * FROM film AS f2, produce AS p2, casa_di_produzione AS cdp2
WHERE cdp1.nome = cdp2.nome AND p2.id_film=f2.id AND f2.generi<>'Western'
AND p2.nome_casa_di_produzione=cdp2.nome)
GROUP BY f1.id;
```

#### **quinta query (inserimento):**

Inserisce un nuovo noleggio.

```
INSERT INTO noleggio ("film_id", "progressivo_dvd", "data_inizio", "cliente_cf") VALUES (42, 2, '2021-03-26', 'V1C2X3Z4L5K6J7H8');
```

#### **sesta query (aggiornamento):**

Aggiorna il numero di telefono di un cliente (sempre identificato dal suo cf).

```
UPDATE cliente SET telefono = 3490022555 WHERE cf = 'B2V4N6M8C1X2Z3L5';
```

#### **settima query (aggiornamento):**

Aggiorna la data di fine di un noleggio. Avviene quando un cliente che ha noleggiato un DVD, lo restituisce.

Nota: questo update fa scattare il trigger!

```
UPDATE noleggio SET data_fine = '2021-04-05' WHERE film_id = 42 AND progressivo_dvd = 2 AND data_inizio = '2021-03-26';
```

#### **ottava query (eliminazione):**

Elimina un film presente nel DB.

```
DELETE FROM film WHERE id="id del film";
```

## **4.4 Trigger**

### **Primo trigger:**

Ogni volta che un DVD viene noleggiato, il trigger incrementa il contatore che tiene conto di quante volte un DVD è stato noleggiato.

```

CREATE OR REPLACE FUNCTION incremento()
RETURNS TRIGGER AS $$
BEGIN
    UPDATE dvd SET numero_di_noleggi = numero_di_noleggi +1
        WHERE dvd.progressivo=new.progressivo_dvd AND dvd.id_film = new.film_id;
RETURN new;

END; $$ language plpgsql;

```

```

CREATE TRIGGER incremento_noleggi
AFTER INSERT
ON noleggio
FOR EACH ROW
EXECUTE PROCEDURE incremento();

```

### Esempio per testare il primo trigger:

```

INSERT INTO noleggio ("film_id", "progressivo_dvd", "data_inizio", "cliente_cf") VALUES (15, 4, '2024-05-03', 'A1S2D3F4G5H6J7K8');

```

### Secondo trigger:

```

CREATE OR REPLACE FUNCTION verifica_ritardo()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.data_inizio + INTERVAL '7 days' < NEW.data_fine
    THEN
        NEW.consegna_fuori_tempo = TRUE;
        RETURN NEW;
    ELSE
        IF NEW.data_inizio + INTERVAL '2 days' > NEW.data_fine
        THEN
            NEW.consegna_fuori_tempo = TRUE;
            RETURN NEW;
        END IF;
    END IF;
    RETURN NEW;
END; $$ language plpgsql;

CREATE TRIGGER ritardo
BEFORE UPDATE
ON noleggio
FOR EACH ROW
EXECUTE PROCEDURE verifica_ritardo();

```

### Esempio per testare il secondo trigger:

```

UPDATE noleggio SET data_fine = '2021-04-05' WHERE film_id = 42 AND progressivo_dvd = 2 AND data_inizio = '2021-03-26';

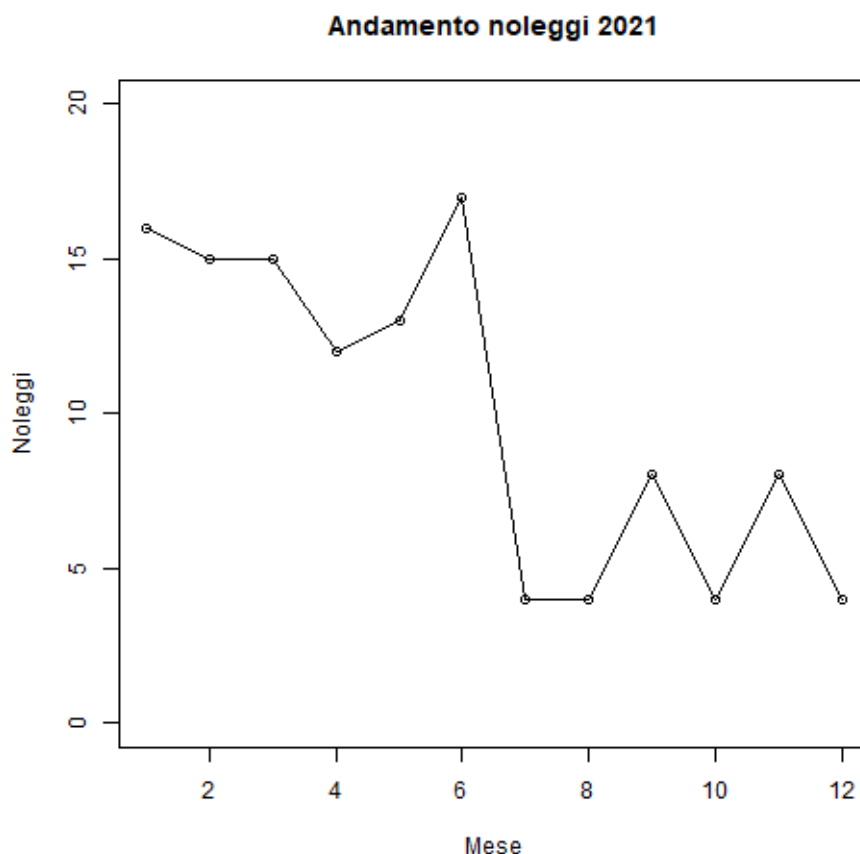
```

## 5 ANALISI DEI DATI CON LINGUAGGIO R

Come parte finale del progetto sono stati analizzati i dati presenti nel DB utilizzando il linguaggio R per creare dei grafici usufruendo della libreria RPostgreSQL.

### 5.1 Andamento dei noleggi per l'anno 2021

Per ricavare l'andamento dei noleggi, sono stati usati i dati presenti nella tabella “noleggio” ed estraendo il mese dal campo “data\_inizio” di tutti i noleggi compresi tra il 1° gennaio ed il 31 dicembre dell'anno 2021, raggruppati per il mese.



NOTA: per estrarre il mese dal campo “data\_inizio” è stata usata la funzione di postgres “date\_part(‘month’, data\_inizio)” per selezionare solo il mese all’interno di un campo di tipo “date”.

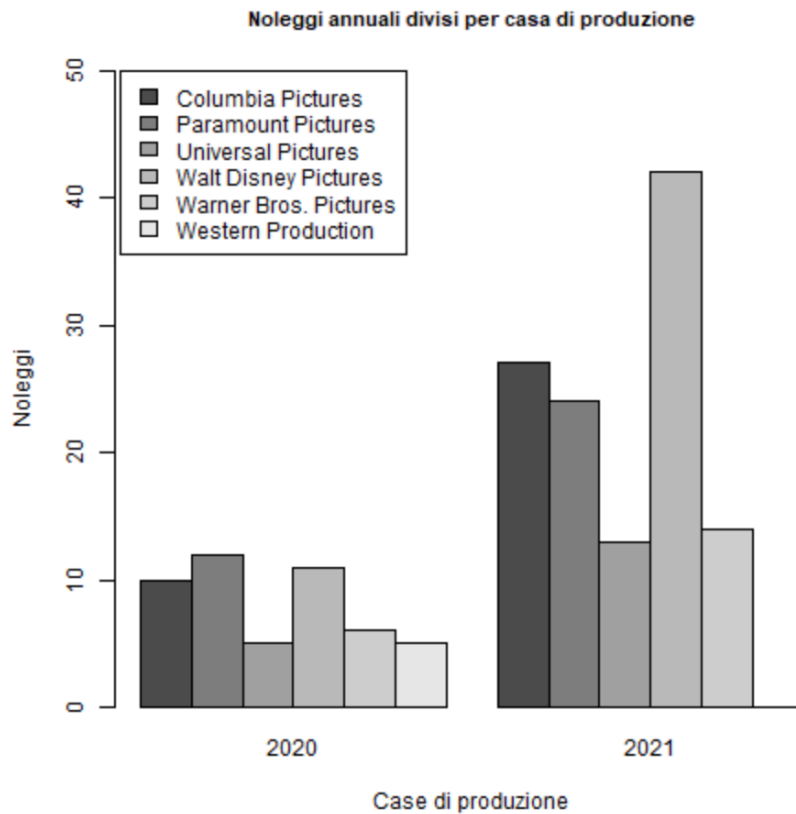
In questo grafico a linee si può notare come nell'anno 2021 sono stati effettuati più noleggi ad inizio anno rispetto alla fine.

### 5.2 Analisi dei noleggi per casa di produzione

Utilizzando i dati presenti nella tabella “noleggio”, “produce” e “casa\_di\_produzione” sono stati ricavati i dati dei noleggi per gli anni 2020 e 2021 e passando per la tabella “produce” sono stati raggruppati per casa di produzione.

Questa rappresentazione in formato di grafico a barre raggruppate fa notare come nel 2020 complessivamente ci siano stati meno noleggi e che la clientela mediamente preferisca noleggiare film prodotti dalla Walt Disney Pictures.

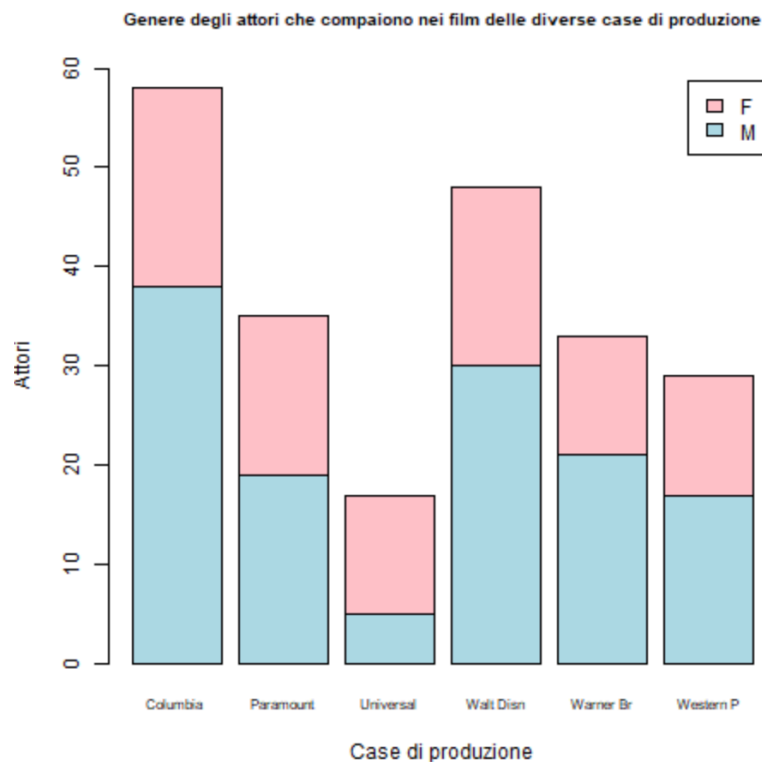
NOTA: per raggruppare le varie case di produzione sono stati utilizzati due JOIN ed un RIGHT JOIN finale per includere le case di produzione a cui non è stato noleggiato nessun film per gli anni analizzati.



### 5.3 Genere degli attori protagonisti

Utilizzando i dati presenti nelle tabelle “attori”, “compare\_in” e “produce” sono stati ricavati i dati relativi al genere degli attori (protagonisti) che compaiono nei film prodotti da una specifica casa di produzione.

Da questo grafico è possibile notare che nei film presenti nel DB ci sia una maggioranza di attori protagonisti di genere maschile.



Nota: i nomi delle case di produzione all'interno del grafico sono state troncate a 9 lettere per rendere più leggibili i label.