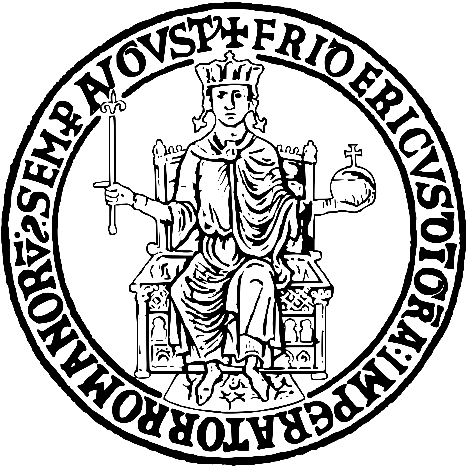
UNIVERSITÀ DEGLI STUDI DI NAPOLI “FEDERICO II”

DIPARTIMENTO DI INGEGNERIA ELETTRICA E TRECNOLOGIE DELL’INFORMAZIONE

****

CORSO DI LAUREA IN INFORMATICA

INSEGNAMENTO DI BASI DI DATI I

ANNO ACCADEMICO 2020/2021

Progettazione e sviluppo di una base di dati relazionale per un sistema di gestione per un multisala

*Autore:**Docenti:*

Crescenzo Lucio CICATIELLO Prof. Adriano PERON

N86003457 Prof. Silvio BARRA

cr.cicatiello@studenti.unina.it

Indice

[**Descrizione del progetto** 3](#_Toc62319016)

[Analisi del problema 3](#_Toc62319017)

[**Cenni teorici** 3](#_Toc62319018)

[UML 3](#_Toc62319019)

[Classi 3](#_Toc62319020)

[Associazioni 3](#_Toc62319021)

[Generalizzazione 4](#_Toc62319022)

[**Progettazione concettuale** 5](#_Toc62319023)

[Class Diagram 5](#_Toc62319024)

[Attributi multipli 5](#_Toc62319025)

[Identificativi 5](#_Toc62319026)

[**Dizionari** 6](#_Toc62319027)

[Dizionario delle classi 6](#_Toc62319028)

[Dizionario delle associazioni 8](#_Toc62319029)

[Dizionario dei vincoli 9](#_Toc62319030)

[**Progettazione logica** 10](#_Toc62319031)

[Schema logico 10](#_Toc62319032)

[Traduzione delle associazioni 11](#_Toc62319033)

[Schema logico 11](#_Toc62319034)

[**Definizioni SQL** 12](#_Toc62319035)

[Definizioni delle tabelle 12](#_Toc62319036)

[Definizione della tabella Attori 12](#_Toc62319037)

[Definizione della tabella Registi 12](#_Toc62319038)

[Definizione della tabella Paesi 12](#_Toc62319039)

[Definizione della tabella Film 13](#_Toc62319040)

[Definizione della tabella Attori\_film 13](#_Toc62319041)

[Definizione della tabella Generi 13](#_Toc62319042)

[Definizione della tabella Genere\_film 14](#_Toc62319043)

[Definizione della tabella Proiezioni 14](#_Toc62319044)

[Definizione della tabella Biglietti 15](#_Toc62319045)

[Definizione della tabella Sale 15](#_Toc62319046)

[Definizione della tabella Sistemi\_audio 15](#_Toc62319047)

[Definizione della tabella Tecnologie\_proiezione 15](#_Toc62319048)

[Definizione delle view 16](#_Toc62319049)

[Definizione dei trigger 17](#_Toc62319050)

# **Descrizione del progetto**

## Analisi del problema

Si sviluppi una base di dati per la gestione di un cinema multisala. Il sistema deve tenere traccia dei film proiettati in ciascuna sala, della schedulazione di ogni spettacolo e delle caratteristiche di ogni sala (audio dolby, tecnologia IMAX). Il sistema dovrà permettere di stimare quali sono le fasce orarie di maggior affluenza, gli spettacoli più remunerativi e le sale maggiormente occupate durante gli orari di maggior affluenza. Il problema verrà riproposto dopo un’introduzione a vari cenni teorici di un class diagram UML.

# **Cenni teorici**

## **UML**

Per la realizzazione di questo progetto è stato necessario l’utilizzo di un diagramma UML. **L’Unified Modeling Language (UML)** è stato realizzato per rappresentare in modo visivo la struttura ,il comportamento e gli oggetti che compongono un sistema software. Grazie all’uso di questo strumento abbiamo una migliore creazione e modellazione di software principalmente orientati agli oggetti.

## **Classi**

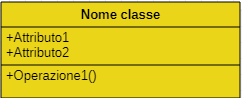
Le **classi** anche dette istanze od entità rappresentato il principale elemento nei diagrammi UML. Ogni classe è composta da degli **attributi**, che descrivono le caratteristiche della classe, e da **operazioni o metodi** che ne derivano il comportamento. Le classi sono rappresentare graficamente come in *Figura 1* In un rettangolo diviso in tre riquadri che contengono rispettivamente nome della classe, attributi ed operazioni.

Figura 1 – Esempio di classe

## **Associazioni**

Le **associazioni** rappresentano le relazioni che intercorrono tra le varie classi di un diagramma.

Vengono rappresentate con una linea che congiunge le due classi sulla quale sono definite alle estremità le molteplicità dell’associazione ed il ruolo dell’associazione (vedi *Figura 2*).

Figura 2 – Rappresentazione di una associazione

Le **molteplicità** possono essere **\*, 1, 0..1, 1..\***.Inoltre l’associazione può essere rappresentata con **una classe collegata con una linea tratteggiata** alla linea di associazione e si possono descrivere le operazioni e gli attributi della classe di associazione.

Le **classi di associazione possono coinvolgere più di due** **classi** tutte con una molteplicità **\*** e vengono rappresentate graficamente con un rombo(vedi *Figura 3*) a cui sono collegate le varie classi con le rispettive linee di associazione.

Inoltre, le associazioni possono specializzarsi e dividersi in:

**Aggregazione**

**Un’aggregazione** (vedi *Figura 3*) specifica una associazione tra un aggregato ed una sua parte costituente che può esiste indipendente dall’aggregato. Per esempio, lo studente di un’aula non dipende dall’aula e viceversa ugual ragionamento è applicabile all’aula. La loro eliminazione non influisce sulle classi ad esse associate.

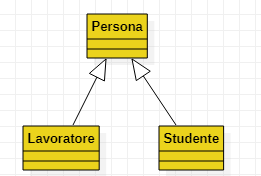
Figura 3 – Rappresentazione di un’aggregazione

**Composizioni**

**Una composizione** (vedi *Figura 4*) identifica invece una forte proprietà tra classi. Per esempio, un’aula che appartiene ad un’università. Se eliminiamo la classe università anche l’aula verrà eliminata di conseguenza.

Figura 4 – Rappresentazione di una composizione

## **Generalizzazione**

Nel diagramma UML, una **relazione di generalizzazione** (vedi *Figura 5*) è una relazione in cui una classe (sottoclasse) dipende da un'altra classe (classe generale) e che assimila da quest’ultima tutti gli attributi, le operazioni e le relazioni della classe principale. La sottoclasse viene definita specializzazione della classe generale , viceversa la classe generale è detta generalizzazione della sottoclasse.

Quando la classe generale è specializzata in varie sottoclassi può essere esclusiva cioè ogni istanza della classe generale deve far parte di una ed una sola delle sottoclassi, mentre nel caso della non esclusività no.

Infine, la generalizzazione può essere totale quindi ogni istanza della classe generale deva far parte almeno ad una sottoclasse mentre in caso della generalizzazione parziale può non appartenere a nessuna delle sottoclassi.

Figura 5 – Generalizzazione

# **Progettazione concettuale**

## Class Diagram

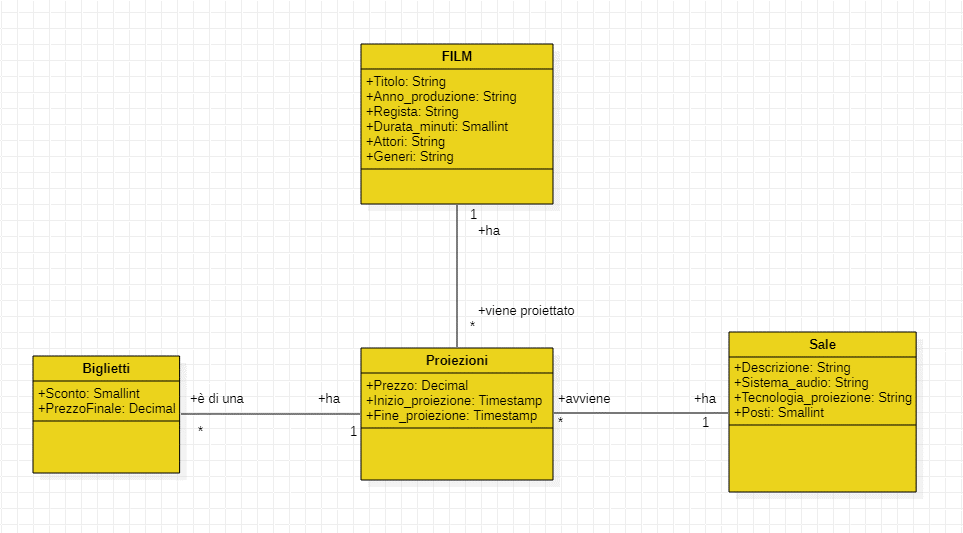
Il seguente **class diagram** in *Figura 6* è una prima versione contente attributi multipli ed enumerazioni che successivamente saranno tramutate in entità per poi portare alla creazione di una seconda versione ristrutturata di quest’ultimo.

Figura 6 - Class Diagram

### **Attributi multipli**

Gli attributi multipli sono stati trasformati in classi (come possiamo notare nella *Figura 7)* a causa della loro molteplicità. I generi ed i film come anche gli attori ed i film sono stati collegati da una relazione molti a molti e dalle rispettive associazioni **genere\_film** e **attori\_film**.

Mentre per gli attributi multipli di **sale** è stato necessario creare delle classi perché non è possibile stimare quali siano o quali saranno i **sistemi audio e tecnologie di proiezione** con il progredire del tempo.

Inoltre, diamo per convenzione che il regista di un film sia al massimo uno.

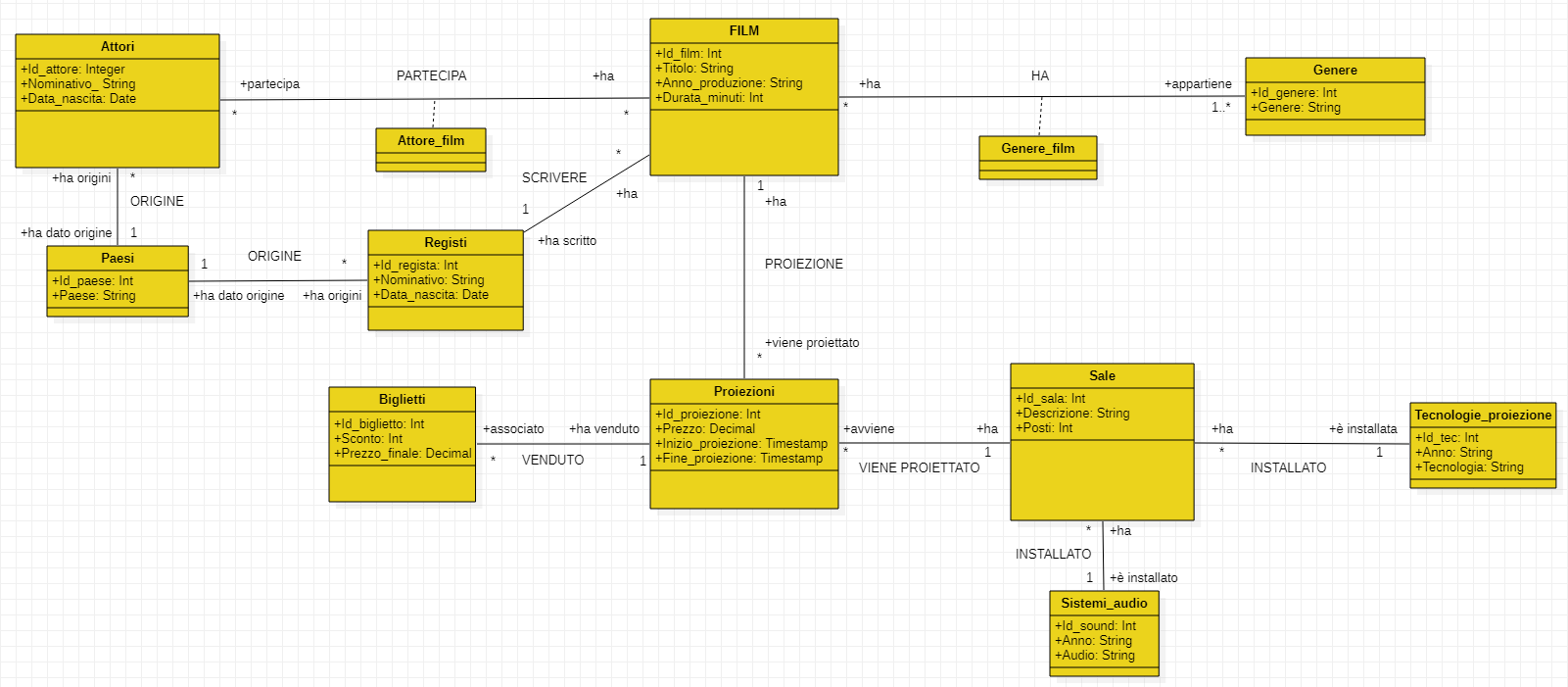


Figura 7- Class Diagram Ristrutturato

La tabella paesi è stata creata per evitare la creazione di un’enumerazione e per aver modo di aggiornare con costanza i paesi col variare del tempo.

### **Identificativi**

Per tutte le classi si è scelto di impiegare chiavi surrogate e quindi per convenzione le chiavi primarie inizieranno con il prefisso Id\_.

# **Dizionari**

## Dizionario delle classi

|  |  |  |
| --- | --- | --- |
| ***Classe*** | ***Descrizione*** | Attributi |
| ***Film*** | Definizione della classe film. | **Id\_film** (*integer*): chiave che identifica univocamente un’istanza della classe film.  **Titolo**(*string*): Nome associato al film.  **Anno\_produzione**(*string*): Anno in cui il film è stato pubblicato.  **Durata\_minuti** (*integer*): Durata del film in minuti. |
| ***Proiezioni*** | Definizione della classe proiezioni. | **Id\_proiezione** (*integer*): chiave che identifica univocamente un’istanza della classe proiezioni.  **Prezzo** (*deciamal*): prezzo della proiezione.  **Inizio\_proiezione** (*timestamp*): data ed ora in cui inizia la proiezione.  **Fine\_proiezione** (*timestamp*): data ed ora in cui termina la proiezione. |
| ***Sale*** | Definizione della classe sale. | **Id\_sala** (*integer*): chiave che identifica univocamente un’istanza della classe sale.  **Descrizione** (*string*): Nome della sala.  **Posti** (*integer*): numero di posti massimi. |
| ***Biglietti*** | Definizione della classe biglietti. | **Id\_biglietto** (*integer*): chiave che identifica univocamente un’istanza della classe biglietti.  **Sconto** (*integer*): sconto da applicare al prezzo della proiezione.  **Prezzo\_finale** (*decimal*): prezzo finale del biglietto scontato. |
| ***Sistemi\_audio*** | Definizione della classe sistemi\_audio. | **Id\_sound** (*integer*): chiave che identifica univocamente un’istanza della classe sistemi\_audio.  **Anno** (*string*): anno di creazione.  **Audio** (*string*): nome della tecnologia. |
| ***Tecnologia\_proiezione*** | Definizione della classe tecnologia\_proiezione. | **Id\_tec** (*integer*): chiave che identifica univocamente un’istanza della classe tecnologia\_proiezione.  **Anno** (*string*): anno di creazione.  **Audio** (*string*): nome della tecnologia. |
| ***Registi*** | Definizione della classe registi. | **Id\_regista** (*integer*): chiave che identifica univocamente un’istanza della classe registi.  **Nominativo** (*string*): Nome e cognome regista.  **Data\_nascita** (*date*): data di nascita. |
| ***Paesi*** | Definizione della classe paesi. | **Id\_regista** (*integer*): chiave che identifica univocamente un’istanza della classe paesi.  **Paese** (*string*): descrizione paese. |
| ***Attori*** | Definizione della classe attori. | **Id\_attore** (*integer*): chiave che identifica univocamente un’istanza della classe attori.  **Nominativo** (*string*): Nome e cognome regista.  **Data\_nascita** (*date*): data di nascita. |
| ***Attore\_film*** | Descrive l’associazione tra attori e film. |  |
| ***Genere\_film*** | Descrive l’associazione tra generi e film. |  |

### Dizionario delle associazioni

|  |  |  |
| --- | --- | --- |
| Associazione | Descrizione | Classi coinvolte |
| Origine(attore-paesi) | Esprime la relazione tra attore e paesi. | **Classe [1]** *ruolo* (**ha origini**): Indica il paese di origine dell’attore.  **Classe [0..\*]** *ruolo* (**ha dato origine**): Indica quali attori ha dato origine un paese. |
| Origine(registi-paesi) | Esprime la relazione tra registi e paesi. | **Classe [1]** *ruolo* (**ha origini**): Indica il paese di origine del regista.  **Classe [0..\*]** *ruolo* (**ha dato origine**): Indica quali registi ha dato origine un paese. |
| Partecipa | Esprime la relazione tra attore e film. | **Classe [1..\*]** *ruolo* (**ha**): Indica quali attori ha un film.  **Classe [1..\*]** *ruolo* (**partecipa**): Indica un attore a quali film ha partecipato. |
| Scrivere | Esprime la relazione tra regista e film. | **Classe [1]** *ruolo* (**ha**): Indica un film da quale regista è stato scritto.  **Classe [1..\*]** *ruolo* (**ha scritto**): Indica un regista quali film ha scritto. |
| Ha | Esprime la relazione tra genere e film. | **Classe [1..\*]** *ruolo* (**ha**): Indica il film quali generi ha.  **Classe [1..\*]** *ruolo* (**appartiene**): Indica il genere a quali film appartiene. |
| Proiezione | Esprime la relazione tra film e proiezioni. | **Classe [1]** *ruolo* (**ha**): Indica quale film viene proiettato.  **Classe [1..\*]** *ruolo* (**viene proiettato**): Indica in quali proiezioni viene proiettato. |
| Venduto | Esprime la relazione tra proiezioni e biglietti. | **Classe [0..\*]** *ruolo* (**ha venduto**): Indica quali biglietti sono stati venduti per una proiezione.  **Classe [1]** *ruolo* (**associato**): Indica a quale proiezione un biglietto è associato. |
| Viene proiettato | Esprime la relazione tra proiezioni e sale. | **Classe [1]** *ruolo* (**avviene**): Indica in quale sala avviene la proiezione.  **Classe [0..\*]** *ruolo* (**ha**): Indica una sala quali proiezioni ha. |
| Installato(tec.proi. -sale) | Esprime la relazione tra sale e tecnologie\_proiezione. | **Classe [1]** *ruolo* (**ha**): Indica una sala quale tecnologia di proiezione ha installata.  **Classe [0..\*]** *ruolo* (**è installata**):Indica in quali sale è installata. |
| Installato(sis.audio-sale) | Esprime la relazione tra sale e sistemi\_audio. | **Classe [1]** *ruolo* (**ha**): Indica una sala quale sistema audio ha installata.  **Classe [0..\*]** *ruolo* (**è installata**):Indica in quali sale è installata. |

### Dizionario dei vincoli

|  |  |  |
| --- | --- | --- |
| ***Vincolo*** | ***Tipo*** | **Descrizione** |
| ***DATA\_NASCITA\_CONCORDE*** | Dominio | **La data di nascita deve essere concorde e quindi minore della data odierna.** |
| ***SOLO\_LETTERE*** | Dominio | **Il nominativo non ammette caratteri numerici.** |
| ***CHECKDURATAMAX*** | Dominio | **Un film non può durare più di 24h.** |
| ***CHECKANNO*** | Dominio | **L’anno di produzione deve essere compreso tra il 1895 e l’anno corrente.** |
| ***CHECKPOSTIMAX*** | Dominio | **Le sale possono essere massimo di 500 posti.** |
| ***MAGGIORE\_ZERO*** | Dominio | **Il prezzo deve essere positivo.** |
| ***ORARI\_CONCORDI*** | N-upla | **Gli orari di inizio e fine proiezione devono essere concordi.** |
| ***INIZIO\_FINE\_NO24H*** | N-upla | **Un film non può durare più di 24h.** |
| ***SCONTO\_0\_100*** | Dominio | **Lo sconto deve esser compreso tra 0% e 100%.** |
| ***MIN\_PROIEZIONE\_CONCORDE\_CON\_MIN\_FILM*** | Interrelazionale | **I minuti di proiezione di un film devono essere concordi con la durata del film stesso.** |
| ***SALA\_PIENA*** | Interrelazionale | **Non possono essere venduti biglietti per una sala piena.** |
| ***PROIEZIONE\_NON\_POSSIBILE*** | Intrarelazionale | **Non possono esistere due proiezioni nella stessa sala e nello stesso momento.** |
| ***SCONTO\_COERENTE\_CON\_PREZZO\_FINALE*** | Interrelazionale | **Il prezzo finale e lo sconto devono essere coerenti con lo sconto applicato al prezzo della proiezione.** |
| ***UNIQUE-PAESE*** | Intrarelazionale | **Non posso esistere due paesi con lo stesso nome.** |
| ***UNIQUE-AUDIO*** | Intrarelazionale | **Non possono esistere due sistemi audio con lo stesso nome.** |
| ***UNIQUE-TECNOLOGIA*** | Intrarelazionale | **Non possono esistere due tecnologie con lo stesso nome.** |
| ***UNIQUE-GENERE*** | Intrarelazionale | **Non possono esistere due generi con lo stesso nome.** |

# **Progettazione logica**

## Schema logico

**Paesi**  (*Id\_paese,paese*)

**Sistemi\_audio** (Id*\_sound,anno,audio*)

**Tecnologie\_proiezione** (*Id\_tec,anno,tecnologia*)

**Generi** (*Id\_genere,genere*)

**Attori** (*Id\_attore,nominativo,paese\_di\_origine,data\_nascita*)

|  |
| --- |
| *Chiavi esterne*: *paese\_di\_origine* → Paesi.Id\_paese |

**Registi** (*Id\_regista,nominativo,paese\_di\_origine,data\_nascita*)

|  |
| --- |
| *Chiavi esterne*: *paese\_di\_origine* → Paesi.Id\_paese |

**Film** (*Id\_film,titolo,anno\_produzione,id\_regista,durata\_minuti,url\_poster*)

|  |
| --- |
| *Chiavi esterne*: Id\_regista → Regista.Id\_regista |

**Attori\_film** (*Id\_attore,Id\_film*)

|  |
| --- |
| *Chiavi* *esterne*: Id\_attore → Attori.Id\_attore  Id\_film → Film.Id\_film |

**Genere\_film** (*Id\_genere,Id\_film*)

|  |
| --- |
| *Chiavi* *esterne*: Id\_genere → Generi.Id\_ genere  Id\_film → Film.Id\_film |

**Sale** (*Id\_sala,descrizione,sistema\_audio,tecnologia\_proiezione,posti*)

|  |
| --- |
| *Chiavi* *esterne*: *sistema\_audio* → Sistemi\_audio.Id\_sound  *tecnologia\_proiezione* → Tecnologie\_proiezione.Id\_tec |

**Proiezioni** (*Id\_proiezione,Id\_sala,Id\_film,prezzo,inizio\_proiezione,fine\_proiezione*)

|  |
| --- |
| *Chiavi* *esterne*: Id\_sala → Sale.Id\_sala  Id\_film → Film.Id\_film |

**Biglietti** (*Id\_proiezione,Id\_biglietto,sconto,prezzo\_finale*)

|  |
| --- |
| *Chiavi* *esterne*: Id\_proiezione → Biglietti.Id\_proiezione |

## Traduzione delle associazioni

|  |  |
| --- | --- |
| ***Associazione*** | ***Implementazione*** |
| ***Ha dato origine(attori)*** | Chiave esterna in Attori → Paesi |
| ***Ha dato origine(registi)*** | Chiave esterna in Registi → Paesi |
| ***Ha scritto*** | Chiave esterna in Film → Registi |
| ***Viene proiettato*** | Chiave esterna in Proiezioni → Film |
| ***Ha venduto*** | Chiave esterna in Biglietti → Proiezioni |
| ***Avviene*** | Chiave esterna in Proiezioni → Sale |
| ***Ha(sale-sistemiaudio)*** | Chiave esterna in Sale → Sistemi\_audio |
| ***Ha(sale-tecnologiafilm)*** | Chiave esterna in Sale → Tecnologia\_film |
| ***Partecipa*** | Chiave esterna in Attore\_film → Attori |
| ***Ha(film-attori)*** | Chiave esterna in Attore\_film → Film |
| ***Ha(film-registi)*** | Chiave esterna in Genere\_film → Film |
| ***Appartiene*** | Chiave esterna in Genere\_film → Generi |

## Schema logico

|  |  |
| --- | --- |
| **Paesi** | (*Id\_paese,paese*) |
| **Sistemi\_audio** | (Id*\_sound,anno,audio*) |
| **Tecnologie\_proiezione** | (*Id\_tec,anno,tecnologia*) |
| **Generi** | (*Id\_genere,genere*) |
| **Attori** | (*Id\_attore,nominativo,paese\_di\_origine,data\_nascita*) |
| **Registi** | (*Id\_regista,nominativo,paese\_di\_origine,data\_nascita*) |
| **Film** | (*Id\_film,titolo,anno\_produzione,id\_regista,durata\_minuti,url\_poster*) |
| **Attori\_film** | (*Id\_attore,Id\_film*) |
| **Genere\_film** | (*Id\_genere,Id\_film*) |
| **Sale** | (*Id\_sala,descrizione,sistema\_audio,tecnologia\_proiezione,posti*) |
| **Proiezioni** | (*Id\_proiezione,Id\_sala,Id\_film,prezzo,inizio\_proiezione,fine\_proiezione*) |
| **Biglietti** | (*Id\_proiezione,Id\_biglietto,sconto,prezzo\_finale*) |

# **Definizioni SQL**

## Definizioni delle tabelle

Per comodità nelle tabelle è stato usato lo pseudo tipo SERIAL che genera una sequenza di interi per utilizzarli come progressivi unici per le istanze delle tabelle. Inoltre, nella parte finale della definizione SQL verranno inseriti i trigger e le varie view utili ad un futuro applicativo.

## Definizione della tabella Attori

**CREATE** **TABLE** ATTORI(

ID\_ATTORE SERIAL **PRIMARY** **KEY**,

NOMINATIVO VARCHAR(50) **NOT** **NULL**,

PAESE\_DI\_ORIGINE VARCHAR(30),

DATA\_NASCITA DATE,

**CONSTRAINT** DATA\_NASCITA\_CONCORDE **CHECK** (DATA\_NASCITA < **CURRENT\_DATE**),

**CONSTRAINT** SOLO\_LETTERE **CHECK** ((NOMINATIVO ~\* '[0-9]') **IS** **FALSE**)

);

## Definizione della tabella Registi

**CREATE** **TABLE** REGISTI(

ID\_REGISTA SERIAL **PRIMARY** **KEY**,

NOMINATIVO VARCHAR(50) **NOT** **NULL**,

PAESE\_DI\_ORIGINE VARCHAR(20) **NOT** **NULL**,

DATA\_NASCITA DATE **NOT** **NULL**,

**CONSTRAINT** DATA\_NASCITA\_CONCORDE **CHECK** (DATA\_NASCITA < **CURRENT\_DATE**),

**CONSTRAINT** SOLO\_LETTERE **CHECK** ((NOMINATIVO ~\* '[0-9]') **IS** **FALSE**)

);

## Definizione della tabella Paesi

**CREATE** **TABLE** PAESI(

ID\_PAESE SMALLSERIAL **PRIMARY** **KEY**,

PAESE VARCHAR(60) **UNIQUE**

);

## Definizione della tabella Film

**CREATE** **TABLE** FILM(

ID\_FILM SERIAL **PRIMARY** **KEY**,

TITOLO VARCHAR(100) **NOT** **NULL**,

ANNO\_PRODUZIONE VARCHAR(4) **NOT** **NULL**,

ID\_REGISTA INTEGER,

DURATA\_MINUTI SMALLINT **NOT** **NULL**,

URL\_POSTER VARCHAR(500),

**CONSTRAINT** CHECKDURATAMAX **CHECK** (DURATA\_MINUTI **BETWEEN** 1 **AND** 1440),

**CONSTRAINT** CHECKANNO **CHECK** (**CAST**(ANNO\_PRODUZIONE **AS** INTEGER) **BETWEEN** 1895 **AND** DATE\_PART('YEAR', **CURRENT\_DATE**)),

**CONSTRAINT** FK\_REGISTA **FOREIGN** **KEY**(ID\_REGISTA) **REFERENCES** REGISTI(ID\_REGISTA)

);

## Definizione della tabella Attori\_film

**CREATE** **TABLE** ATTORI\_FILM(

ID\_ATTORE INTEGER,

ID\_FILM INTEGER,

**CONSTRAINT** PK\_ATTORI\_FILM **PRIMARY** **KEY**(ID\_ATTORE, ID\_FILM), **CONSTRAINT** FK\_ATTOREFILM **FOREIGN** **KEY**(ID\_ATTORE) **REFERENCE S** ATTORI(ID\_ATTORE),

**CONSTRAINT** FK\_FILMATTORE **FOREIGN** **KEY**(ID\_FILM) **REFERENCES** FILM(ID\_FILM)

);

## Definizione della tabella Generi

**CREATE** **TABLE** GENERI(

ID\_GENERE SMALLSERIAL **PRIMARY** **KEY**,

GENERE VARCHAR(20) **UNIQUE**

);

## Definizione della tabella Genere\_film

**CREATE** **TABLE** GENERE\_FILM(

ID\_GENERE INTEGER,

ID\_FILM INTEGER,

**CONSTRAINT** PK\_GENERE\_FILM **PRIMARY** **KEY**(ID\_GENERE, ID\_FILM),

**CONSTRAINT** FK\_GENEREFILM **FOREIGN** **KEY**(ID\_GENERE) **REFERENCES** GENERI(ID\_GENERE),

**CONSTRAINT** FK\_FILMGENERE **FOREIGN** **KEY**(ID\_FILM) **REFERENCES** FILM(ID\_FILM)

);

## Definizione della tabella Proiezioni

**CREATE** **TABLE** PROIEZIONI(

ID\_PROIEZIONE SERIAL **PRIMARY** **KEY**,

ID\_SALA INTEGER,

ID\_FILM INTEGER,

PREZZO DECIMAL **NOT** **NULL**,

INIZIO\_PROIEZIONE **TIMESTAMP** **NOT** **NULL**,

FINE\_PROIEZIONE **TIMESTAMP** **NOT** **NULL**,

**CONSTRAINT** MAGGIORE\_ZERO **CHECK** (PREZZO > 0),

**CONSTRAINT** FK\_SALA **FOREIGN** **KEY**(ID\_SALA) **REFERENCES** SALE(ID\_SALA),

**CONSTRAINT** FK\_FILM **FOREIGN** **KEY**(ID\_FILM) **REFERENCES** FILM(ID\_FILM) **ON** **DELETE** **CASCADE**,

**CONSTRAINT** ORARI\_CONCORDI **CHECK** (INIZIO\_PROIEZIONE < FINE\_PROIEZIONE),

**CONSTRAINT** INIZIO\_FINE\_NO24H **CHECK** ((DATE\_PART('day', FINE\_PROIEZIONE::**TIMESTAMP** - INIZIO\_PROIEZIONE::**TIMESTAMP**) \* 24 + DATE\_PART('hour', FINE\_PROIEZIONE::**TIMESTAMP** - INIZIO\_PROIEZIONE::**TIMESTAMP**))<=24)

);

## Definizione della tabella Biglietti

**CREATE** **TABLE** BIGLIETTI(

ID\_PROIEZIONE INTEGER,

ID\_BIGLIETTO SERIAL **PRIMARY** **KEY**,

SCONTO SMALLINT,

PREZZOFINALE DECIMAL **NOT** **NULL**, **CONSTRAINT** SCONTO\_0\_100 **CHECK** (SCONTO **BETWEEN** 0 **AND** 100), **CONSTRAINT** FK\_PROIEZIONE **FOREIGN** **KEY**(ID\_PROIEZIONE) **REFERENCES** PROIEZIONI(ID\_PROIEZIONE) **ON** **DELETE** **CASCADE**

);

## Definizione della tabella Sale

**CREATE** **TABLE** SALE(

ID\_SALA SERIAL **PRIMARY** **KEY**,

DESCRIZIONE VARCHAR(20) **NOT** **NULL**,

SISTEMA\_AUDIO INTEGER,

TECNOLOGIA\_PROIEZIONE INTEGER,

POSTI SMALLINT **NOT** **NULL**, **CONSTRAINT** CHECKPOSTIMAX **CHECK** (POSTI **BETWEEN** 1 **AND** 500)

);

## Definizione della tabella Sistemi\_audio

**CREATE** **TABLE** SISTEMI\_AUDIO(

ID\_SOUND SMALLSERIAL **PRIMARY** **KEY**,

ANNO VARCHAR(4),

AUDIO VARCHAR(50) **UNIQUE**

);

## Definizione della tabella Tecnologie\_proiezione

**CREATE** **TABLE** TECNOLOGIE\_PROIEZIONE(

ID\_TEC SMALLSERIAL **PRIMARY** **KEY**,

ANNO VARCHAR(4),

TECNOLOGIA VARCHAR(50) **UNIQUE**

);

## Definizione delle view

Le view definite servono rispettivamente a permettere di stimare quali sono le fasce orarie di maggior affluenza, gli spettacoli più remunerativi e le sale maggiormente occupate durante gli orari di maggior affluenza.

**CREATE** **VIEW** SPETTACOLI\_RENUMERATIVI **AS**

(**SELECT**

(**SELECT** F.TITOLO

**FROM** FILM F

**WHERE** F.ID\_FILM=P.ID\_FILM)FILM,

**COUNT**(B.ID\_BIGLIETTO)BIGLIETTI\_VENDUTI,

**SUM**(B.PREZZOFINALE)GUADAGNO\_TOTALE

**FROM** BIGLIETTI B

**JOIN** PROIEZIONI P **ON** B.ID\_PROIEZIONE=P.ID\_PROIEZIONE

**GROUP** **BY** FILM

**ORDER** **BY** BIGLIETTI\_VENDUTI,GUADAGNO\_TOTALE

);

**CREATE** **VIEW** ORARI\_MAGGIORE\_AFFLUENZA **AS**

(**SELECT** **COUNT**(\*)AFFLUENZA,

**CAST**(INIZIO\_PROIEZIONE **AS** TIME)FASCIAINIZIO,

**CAST**(FINE\_PROIEZIONE **AS** TIME)FASCIAFINE

**FROM** BIGLIETTI B

**JOIN** PROIEZIONI P **ON** B.ID\_PROIEZIONE=P.ID\_PROIEZIONE

**GROUP** **BY** FASCIAINIZIO, FASCIAFINE

**ORDER** **BY** AFFLUENZA **DESC**

);

**CREATE** **VIEW** AFFLUENZA\_SALE\_ORARI\_MAX **AS**

(**SELECT** **COUNT**(\*)CONTA,

P.ID\_SALA,

**CAST**(P.INIZIO\_PROIEZIONE **AS** TIME)FASCIAINIZIO,

**CAST**(P.FINE\_PROIEZIONE **AS** TIME)FASCIAFINE

**FROM** BIGLIETTI B

**JOIN** PROIEZIONI P **ON** B.ID\_PROIEZIONE=P.ID\_PROIEZIONE

**GROUP** **BY** 2,3,4

**HAVING** **CAST**(P.INIZIO\_PROIEZIONE **AS** TIME)||'-'||

**CAST**(P.FINE\_PROIEZIONE **AS** TIME) **IN** (**SELECT** O.FASCIAINIZIO||'-'

||O.FASCIAFINE **FROM** ORARI\_MAGGIORE\_AFFLUENZA O

**WHERE** O.AFFLUENZA=(**SELECT** **MAX**(O.AFFLUENZA)

**FROM** ORARI\_MAGGIORE\_AFFLUENZA O))

**ORDER** **BY** CONTA **DESC**

);

## Definizione dei trigger

Questi sono i vincoli interrelazionale mostrati nel paragrafo vincoli.

**CREATE** **OR** **REPLACE** **FUNCTION** CONTROLLO\_SALA\_PIENA() **RETURNS** **TRIGGER** **AS** $$

**DECLARE**

POSTI\_SALA INT;

BIGLIETTI\_VENDUTI INT;

**BEGIN**

**SELECT** S.POSTI **INTO** POSTI\_SALA **FROM** SALE S **WHERE** S.ID\_SALA=(**SELECT** P.ID\_SALA **FROM** PROIEZIONI P **WHERE** P.ID\_PROIEZIONE=**NEW**.ID\_PROIEZIONE);

**SELECT** **COUNT**(\*) **INTO** BIGLIETTI\_VENDUTI **FROM** BIGLIETTI B **WHERE**

B.ID\_PROIEZIONE=**NEW**.ID\_PROIEZIONE;

**IF** (POSTI\_SALA = BIGLIETTI\_VENDUTI) **THEN**

RAISE **EXCEPTION** 'SALA PIENA';

**END** **IF**;

**RETURN** **NEW**;

**END**;

$$ **LANGUAGE** PLPGSQL;

**CREATE** **OR** **REPLACE** **FUNCTION** CONTROLLO\_SALA\_FILM\_ORARIO() **RETURNS** **TRIGGER** **AS** $$

**DECLARE**

CONTA INT := 0;

RIGA PROIEZIONI%ROWTYPE;

**BEGIN**

**FOR** RIGA **IN** **SELECT** \* **FROM** PROIEZIONI P

**WHERE** P.ID\_SALA=**NEW**.ID\_SALA **AND**

(P.INIZIO\_PROIEZIONE::DATE **IN** (P.INIZIO\_PROIEZIONE::DATE,P.FINE\_PROIEZIONE::DATE) **OR**

P.FINE\_PROIEZIONE::DATE **IN** (P.INIZIO\_PROIEZIONE::DATE,P.FINE\_PROIEZIONE::DATE))

LOOP

**IF**((**NEW**.INIZIO\_PROIEZIONE **BETWEEN** RIGA.INIZIO\_PROIEZIONE **AND** RIGA.FINE\_PROIEZIONE)

**OR** (**NEW**.FINE\_PROIEZIONE **BETWEEN** RIGA.INIZIO\_PROIEZIONE **AND** RIGA.FINE\_PROIEZIONE)) **THEN**

CONTA := CONTA + 1;

**END** **IF**;

EXIT **WHEN** CONTA > 0;

**END** LOOP;

**IF**(CONTA > 0) **THEN**

RAISE **EXCEPTION** 'SALA GIÀ OCCUPATA PER LA DATA E L''ORARIO INSERITO';

**END** **IF**;

**RETURN** **NEW**;

**END**;

$$ **LANGUAGE** PLPGSQL;

**CREATE** **OR** **REPLACE** **FUNCTION** CONTROLLO\_PROIEZIONE\_DURATA\_FILM() **RETURNS** **TRIGGER** **AS** $$

**DECLARE**

MINUTI\_FILM INT := 0;

**BEGIN**

**SELECT** F.DURATA\_MINUTI **INTO** MINUTI\_FILM **FROM** FILM F **WHERE** F.ID\_FILM=**NEW**.ID\_FILM;

**IF**(((DATE\_PART('DAY', **CAST**(**NEW**.FINE\_PROIEZIONE **AS** **timestamp** ) - **CAST**(**NEW**.INIZIO\_PROIEZIONE **AS** **timestamp** )) \* 24 +

DATE\_PART('HOUR', **CAST**(**NEW**.FINE\_PROIEZIONE **AS** **timestamp** ) - **CAST**(**NEW**.INIZIO\_PROIEZIONE **AS** **timestamp** ))) \* 60 +

DATE\_PART('MINUTE', **CAST**(**NEW**.FINE\_PROIEZIONE **AS** **timestamp** ) - **CAST**(**NEW**.INIZIO\_PROIEZIONE **AS** **timestamp** ))) < MINUTI\_FILM) **THEN**

RAISE **EXCEPTION** 'LA DURATA DEL FILM É MAGGIORE RISPETTO ALLA DURATA DELLA PROIEZIONE ';

**END** **IF**;

**RETURN** **NEW**;

**END**;

$$ **LANGUAGE** PLPGSQL;

*/\*CONTROLLO BIGLIETTI\*/*

**CREATE** **OR** **REPLACE** **FUNCTION** CONTROLLO\_SCONTO\_PREZZO\_FINALE() **RETURNS** **TRIGGER** **AS** $$

**DECLARE**

PREZZO\_PROIEZIONE INT := 0;

**BEGIN**

**SELECT** P.PREZZO **INTO** PREZZO\_PROIEZIONE **FROM** PROIEZIONI P **WHERE** P.ID\_PROIEZIONE=**NEW**.ID\_PROIEZIONE;

**IF**(PREZZO\_PROIEZIONE-(PREZZO\_PROIEZIONE\***NEW**.SCONTO)/100 <> **NEW**.PREZZOFINALE) **THEN**

RAISE **EXCEPTION** 'PREZZOFINALE BIGLIETTO NON COERENTE CON SCONTO APPLICATO';

**END** **IF**;

**RETURN** **NEW**;

**END**;

$$ **LANGUAGE** PLPGSQL;

**CREATE** **TRIGGER** MIN\_PROIEZIONE\_CONCORDE\_CON\_MIN\_FILM

**BEFORE**

**INSERT** **ON** PROIEZIONI

**FOR** **EACH** **ROW** **EXECUTE** **FUNCTION** CONTROLLO\_PROIEZIONE\_DURATA\_FILM();

**CREATE** **TRIGGER** SALA\_PIENA

**BEFORE**

**INSERT** **ON** BIGLIETTI

**FOR** **EACH** **ROW** **EXECUTE** **FUNCTION** CONTROLLO\_SALA\_PIENA();

**CREATE** **TRIGGER** PROIEZIONE\_NON\_POSSIBILE

**BEFORE**

**INSERT** **ON** PROIEZIONI

**FOR** **EACH** **ROW** **EXECUTE** **FUNCTION** CONTROLLO\_SALA\_FILM\_ORARIO();

**CREATE** **TRIGGER** SCONTO\_COERENTE\_CON\_PREZZO\_FINALE

**BEFORE**

**INSERT** **ON** BIGLIETTI

**FOR** **EACH** **ROW** **EXECUTE** **FUNCTION** CONTROLLO\_SCONTO\_PREZZO\_FINALE();