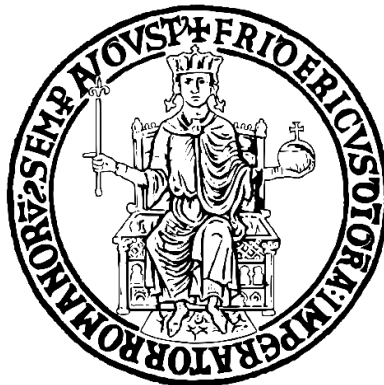


UNIVERSITÀ DEGLI STUDI DI NAPOLI “FEDERICO II”
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE
DELL’INFORMAZIONE



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:

Crescenzo Lucio CICATIELLO
N86003457
cr.cicatiello@studenti.unina.it

Docenti:

Prof. Adriano PERON
Prof. Silvio BARRA

Indice

Descrizione del progetto	3
Analisi del problema.....	3
Cenni teorici	3
UML	3
Classi	3
Associazioni	3
Generalizzazione	4
Progettazione concettuale	5
Class Diagram	5
Attributi multipli.....	5
Identificativi	5
Dizionari	6
Dizionario delle classi	6
Dizionario delle associazioni.....	8
Dizionario dei vincoli	9
Progettazione logica	10
Schema logico	10
Traduzione.....	10
Traduzione delle associazioni.....	10
Schema logico	10
Definizioni SQL	10
Definizione della tabella Attori	10
Definizione della tabella Registi.....	10
Definizione della tabella Paesi	10
Definizione della tabella Film	10
Definizione della tabella Attore_film	10
Definizione della tabella Genere	10
Definizione della tabella Genere_film.....	10
Definizione della tabella Proiezioni	10
Definizione della tabella Biglietti.....	10
Definizione della tabella Sale	10
Definizione della tabella Sistemi_audio	10
Definizione della tabella Tecnologie_proiezione	11

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 orientato agli oggetti.

Classi

Le **classi** anche dette istanze od entità rappresentano 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 rappresentate graficamente come in *Figura 1* In un rettangolo diviso in tre riquadri che contengono rispettivamente nome della classe, attributi ed operazioni.

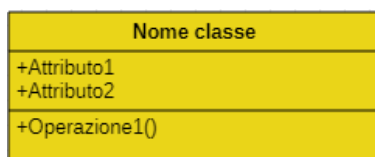


Figura 1 – Esempio di classe

Associazioni

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



Figura 2 – Rappresentazione di una associazione

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*).

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



Figura 3 – Rappresentazione di un'aggregazione

Un'**aggregazione** (vedi *Figura 3*) specifica una associazione tra un aggregato ed una sua parte costituente che può esistere 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.

Composizioni



Figura 4 – Rappresentazione di una composizione

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.

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.

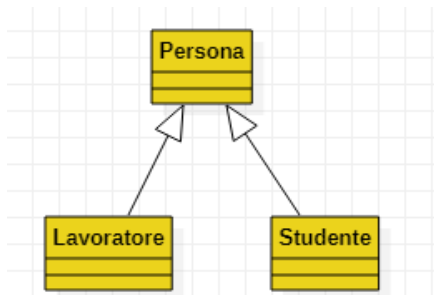


Figura 5 – Generalizzazione

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.

Progettazione concettuale

Class Diagram

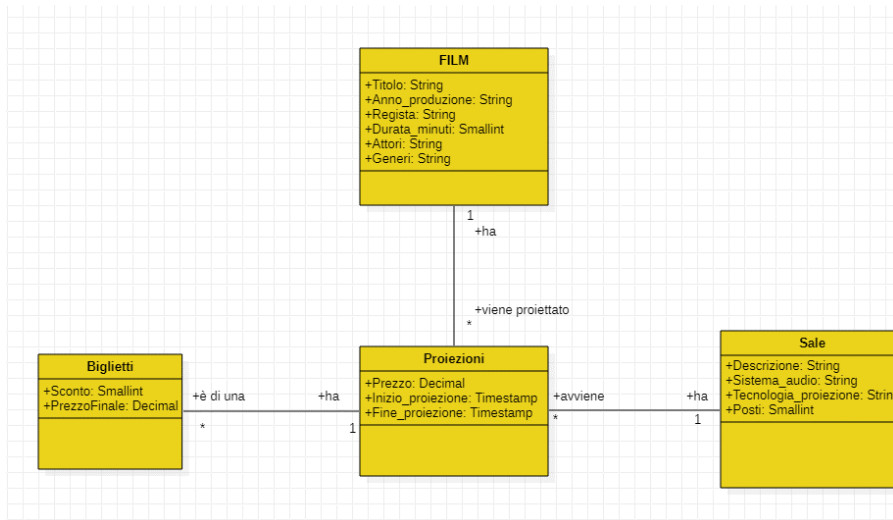


Figura 6 - Class Diagram

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

Attributi multipli

Gli attributi multipli della classe film come **generi** e **attori** 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.

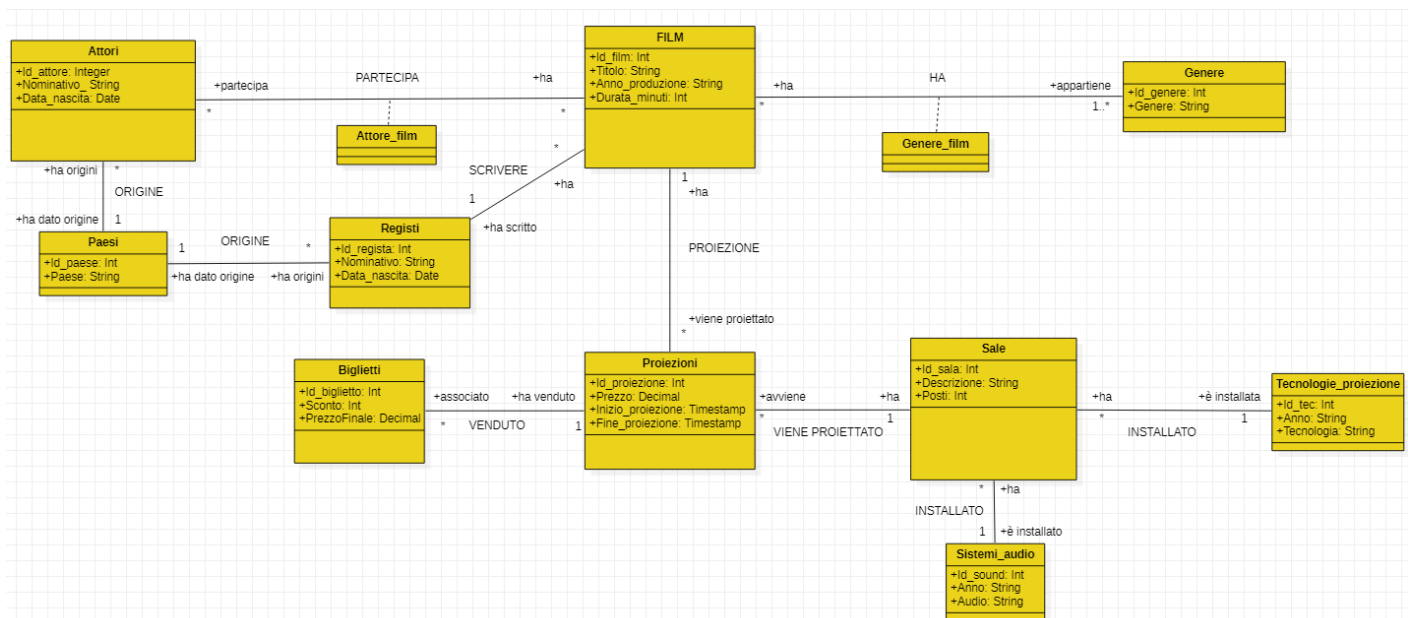


Figura 7- Class Diagram Ristrutturato

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 (<i>integer</i>): chiave che identifica univocamente un'istanza della classe film. Titolo (<i>string</i>): Nome associato al film. Anno_produzione (<i>string</i>): Anno in cui il film è stato pubblicato. Durata_minuti (<i>integer</i>): Durata del film in minuti.
Proiezioni	Definizione della classe proiezioni.	Id_proiezione (<i>integer</i>): chiave che identifica univocamente un'istanza della classe proiezioni. Prezzo (<i>deciamal</i>): prezzo della proiezione. Inizio_proiezione (<i>timestamp</i>): data ed ora in cui inizia la proiezione. Fine_proiezione (<i>timestamp</i>): data ed ora in cui termina la proiezione.
Sale	Definizione della classe sale.	Id_sala (<i>integer</i>): chiave che identifica univocamente un'istanza della classe sale. Descrizione (<i>string</i>): Nome della sala. Posti (<i>integer</i>): numero di posti massimi.
Biglietti	Definizione della classe biglietti.	Id_biglietto (<i>integer</i>): chiave che identifica univocamente un'istanza della classe biglietti. Sconto (<i>integer</i>): sconto da applicare al prezzo della proiezione. Prezzofinale (<i>decimal</i>): prezzo finale del biglietto scontato.
Sistemi_audio	Definizione della classe sistemi_audio.	Id_sound (<i>integer</i>): chiave che identifica univocamente un'istanza della classe sistemi_audio. Anno (<i>string</i>): anno di creazione. Audio (<i>string</i>): nome della tecnologia.

<i>Tecnologia_proiezione</i>	Definizione della classe tecnologia_proiezione.	Id_tec (<i>integer</i>): chiave che identifica univocamente un'istanza della classe tecnologia_proiezione. Anno (<i>string</i>): anno di creazione. Audio (<i>string</i>): nome della tecnologia.
<i>Registi</i>	Definizione della classe registi.	Id_regista (<i>integer</i>): chiave che identifica univocamente un'istanza della classe registi. Nominativo (<i>string</i>): Nome e cognome regista. Data_nascita (<i>date</i>): data di nascita.
<i>Paesi</i>	Definizione della classe paesi.	Id_regista (<i>integer</i>): chiave che identifica univocamente un'istanza della classe paesi. Paese (<i>string</i>): descrizione paese.
<i>Attori</i>	Definizione della classe attori.	Id_attore (<i>integer</i>): chiave che identifica univocamente un'istanza della classe attori. Nominativo (<i>string</i>): Nome e cognome regista. Data_nascita (<i>date</i>): data di nascita.
<i>Attore_film</i>	Descrive l'associazione tra attori e film.	
<i>Genere_film</i>	Descrive l'associazione tra generi e film.	

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 a 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 a quali proiezioni ha.
Installato(tec.proi. -sale)	Esprime la relazione tra sale e tecnologie_proiezione.	Classe [1] ruolo (ha): Indica una sala a 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 a quale sistema audio ha installata. Classe [0..*] ruolo (è installata): Indica in quali sale è installata.

<i>Vincolo</i>	<i>Tipo</i>	<i>Descrizione</i>
<i>DATA_NASCITA_CONCORDE</i>	Dominio	La data di nascita deve essere concorde e quindi minore della data odierna.
<i>SOLO_LETTERE</i>	Dominio	Il nominativo non ammette caratteri numerici.
<i>CHECKDURATAMAX</i>	Dominio	Un film non può durare più di 24h.
<i>CHECKANNO</i>	Dominio	L'anno di produzione deve essere compreso tra il 1895 e l'anno corrente.
<i>CHECKPOSTIMAX</i>	Dominio	Le sale possono essere massimo di 500 posti.
<i>MAGGIORE_ZERO</i>	Dominio	Il prezzo deve essere positivo.
<i>ORARI_CONCORDI</i>	N-upla	Gli orari di inizio e fine proiezione devono essere concordi.
<i>INIZIO_FINE_NO24H</i>	N-upla	Un film non può durare più di 24h.
<i>SCONTO_0_100</i>	Dominio	Lo sconto deve esser compreso tra 0% e 100%.
<i>MIN_PROIEZIONE_CONCORDE_CON_MIN_FILM</i>	Interrelazionale	I minuti di proiezione di un film devono essere concordi con la durata del film stesso.
<i>SALA_PIENA</i>	Interrelazionale	Non possono essere venduti biglietti per una sala piena.
<i>PROIEZIONE_NON_POSSIBILE</i>	Intrarelazionale	Non possono esistere due proiezioni nella stessa sala e nello stesso momento.
<i>SCONTO_COERENTE_CON_PREZZO_FINALI</i>	Interrelazionale	Il prezzo finale e lo sconto devono essere coerenti con lo sconto applicato al prezzo della proiezione.
<i>UNIQUE-PAESE</i>	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

Traduzione

Traduzione delle associazioni

Schema logico

Definizioni SQL

Definizione della tabella Attori

Definizione della tabella Registi

Definizione della tabella Paesi

Definizione della tabella Film

Definizione della tabella Attore_film

Definizione della tabella Genere

Definizione della tabella Genere_film

Definizione della tabella Proiezioni

Definizione della tabella Biglietti

Definizione della tabella Sale

Definizione della tabella Sistemi_audio

