



Università degli Studi di Salerno
Corso di Ingegneria del Software

CineNow
Database Design Document
Versione 1.0



Data: 24/11/2024

Progetto: CineNow	Versione: 1.0
Documento: Database Design Document	Data:22/11/2024

Coordinatore del progetto:

Nome	Matricola

Partecipanti:

Nome	Matricola
Emanuele Iovane	0512120565
Armando Vigliotti	0512117739
Antonio Caiazzo	0512117751

Scritto da:	Caiazzo Antonio, Iovane Emanuele, Vigliotti Armando
--------------------	---

Revision History

Data	Versione	Descrizione	Autore
22/11/2024	0.1	Creazione del Database Design Document	Tutto il team
24/11/2024	1.0	Revisione finale del documento e primo rilascio	Tutto il team

Progetto: CineNow	Versione: 1.0
Documento: Database Design Document	Data:22/11/2024

Indice

1. INTRODUZIONE	4
2. MODELLO CONCETTUALE DEL DATABASE	5
3. SCHEMA RELAZIONALE DEL DATABASE	6
4. STRUTTURA TABELLARE DEL DATABASE	7

Progetto: CineNow	Versione: 1.0
Documento: Database Design Document	Data: 22/11/2024

1. INTRODUZIONE

Questo documento descrive la progettazione dettagliata del database per il sistema CineNow. Include il modello concettuale e lo schema relazionale per supportare la gestione di tutte le funzionalità chiave del sistema, come la gestione degli utenti, la prenotazione dei biglietti, la gestione dei film e delle sale, e il monitoraggio della disponibilità dei posti.

Per gestire la persistenza dei dati, si è scelto di utilizzare un database relazionale, gestendo tramite il DBMS MySQL. La scelta di un database relazionale per questo sistema, è stata definita perché il suo modello tabellare strutturato è ideale per gestire dati, organizzati in modo chiaro e definito. La capacità di supportare le proprietà ACID garantisce operazioni sicure e consistenti, fondamentali per la gestione affidabile delle transazioni. Inoltre, i vincoli di integrità, attraverso l'uso di chiavi primarie ed esterne, permettono di mantenere un alto livello di qualità e coerenza dei dati, riducendo la necessità di controlli complessi a livello applicativo. Inoltre l'utilizzo di query SQL consente interrogazioni avanzate e aggregazioni complesse, rendendo questo approccio estremamente versatile. È stato scelto MySQL come DBMS perché unisce affidabilità e facilità di utilizzo, nonché alta compatibilità con molteplici piattaforme e linguaggi, tra cui JAVA.

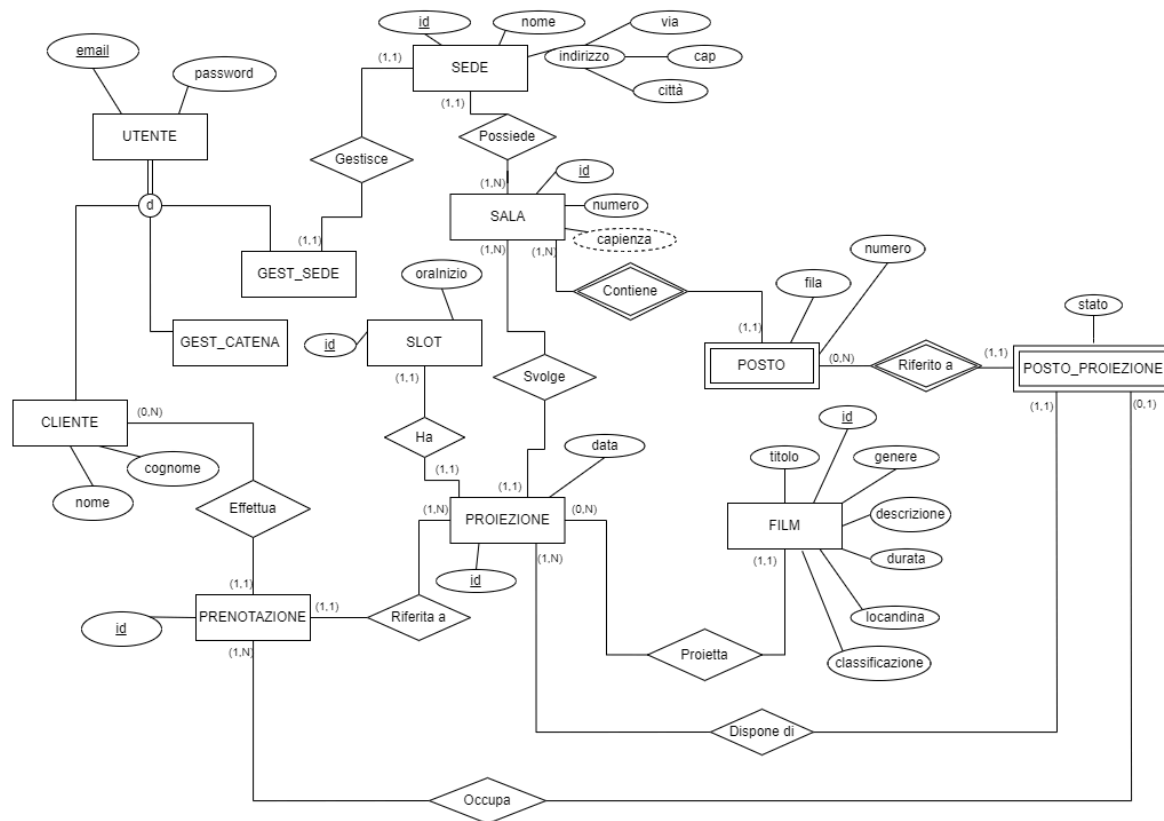
1.1. Struttura del documento

Questo documento è suddiviso in diverse parti per fornire una visione chiara e dettagliata della progettazione del database per CineNow:

- La **prima parte** introduce il contesto e gli obiettivi del database, spiegando le motivazioni delle scelte progettuali.
- La **seconda parte** descrive il **modello concettuale**, includendo il diagramma ER.
- La **terza parte** illustra lo **schema relazionale**, fornendo una mappatura chiara tra il modello concettuale e la struttura tabellare del database.
- La **quarta parte** dettaglia la **struttura tabellare**, includendo i tipi di dati, i vincoli di integrità e le chiavi primarie ed esterne per ogni tabella.

Progetto: CineNow	Versione: 1.0
Documento: Database Design Document	Data: 22/11/2024

2. MODELLO CONCETTUALE DEL DATABASE



Per il modello EER definito, l'attributo “*capienza*”, è definito dal numero di posti di una sala. Si è scelto per chiarezza di definire l'entità “*POSTO_PROIEZIONE*”, anziché mantenere solo la relazione, per semplificare le richieste al database. Lo “*SLOT*” modella gli slot orari disponibili per effettuare le proiezioni. Essi saranno fissi per il nostro sistema, e andranno dalle 18:00 alle 21:30.

Progetto: CineNow	Versione: 1.0
Documento: Database Design Document	Data:22/11/2024

3. SCHEMA RELAZIONALE DEL DATABASE

Dopo la ristrutturazione del modello EER, si è ottenuto il seguente schema logico.

<u>Entità/Relazione</u>	<u>Attributi</u>	<u>Chiave primaria</u>	<u>Chiave esterna</u>
UTENTE	email, password	email	
CLIENTE	UTENTE.email, nome, cognome	UTENTE.email	UTENTE.email
GEST_SEDE	UTENTE.email, SEDE.id	UTENTE.email	UTENTE.email, SEDE.id
GEST_CATENA	UTENTE.email	UTENTE.email	UTENTE.email
SEDE	id, nome, via, cap, città	id	
SALA	id, SEDE.id, numeroSala, capienza	id	SEDE.id
POSTO	SALA.id, fila, numero	SALA.id, fila, numero	SALA.id
PROIEZIONE	id, FILM.id, SALA.id, SLOT.id, data	id	FILM.id, SALA.id, SLOT.id
POSTO_PROIEZIONE	SALA.id, POSTO.fila, POSTO.numero, PROIEZIONE.id, stato	SALA.id, POSTO.fila, POSTO.numero, PROIEZIONE.id	SALA.id, POSTO.fila, POSTO.numero, PROIEZIONE.id
FILM	id, titolo, genere, classificazione, durata, descrizione, locandina	id	id
SLOT	id, oraInizio	id	
PRENOTAZIONE	id, CLIENTE.email, PROIEZIONE.id	id	CLIENTE.email, PROIEZIONE.id
Occupa {PRENOTAZIONE - POSTO_PROIEZIONE}	SALA.id, POSTO.fila, POSTO.numero, PROIEZIONE.id, PRENOTAZIONE.id	SALA.id, POSTO.fila, POSTO.numero, PROIEZIONE.id, PRENOTAZIONE.id	SALA.id, POSTO.fila, POSTO.numero, PROIEZIONE.id, PRENOTAZIONE.id

NOTE TABELLA: Il formato ENTITÀ.attributo, indica l'attributo dell'entità posseditrice.
Le relazioni seguono il seguente formato:
Relazione{ ENTITÀCOINVOLTA1 - ENTITÀCOINVOLTA2 }

Progetto: CineNow	Versione: 1.0
Documento: Database Design Document	Data:22/11/2024

4. STRUTTURA TABELLARE DEL DATABASE

Si indica con PK la chiave primaria, con FK la chiave esterna, con PKC una chiave candidata che fa parte della chiave primaria composta, con il formato ENTITA.attributo, il riferimento all'attributo della specifica ENTITA. Si indica con DEFAULT(valore), il valore di default dell'attributo.

UTENTE		
Attributo	Tipo	Vincoli
email	string	PK
password	string	not null

CLIENTE		
Attributo	Tipo	Vincoli
email	string	PK & FK(UTENTE.email)
nome	string	not null
cognome	sting	not null

GEST_CATENA		
Attributo	Tipo	Vincoli
email	string	PK & FK(UTENTE.email)

GEST_SEDE		
Attributo	Tipo	Vincoli
email	string	PK & FK(UTENTE.email)
id_sede	integer	FK(SEDE.id)

Progetto: CineNow	Versione: 1.0
Documento: Database Design Document	Data:22/11/2024

SEDE		
Attributo	Tipo	Vincoli
id	integer	PK
nome	string	not null
via	string	not null
città	string	not null
cap	string(5 caratteri)	not null

SALA		
Attributo	Tipo	Vincoli
id	integer	PK
id_sede	integer	FK(SEDE.id)
numero	integer	not null
capienza	integer	not null

POSTO		
Attributo	Tipo	Vincoli
id_sala	integer	PKC & FK(SALA.id)
numero	integer	PKC
fila	string	PKC

Progetto: CineNow	Versione: 1.0
Documento: Database Design Document	Data:22/11/2024

FILM		
Attributo	Tipo	Vincoli
id	string	PK
titolo	string	not null
genere	string	not null
classificazione	string	not null
durata	integer	not null
locandina	string	not null
descrizione	string	not null

SLOT		
Attributo	Tipo	Vincoli
id	integer	PK
ora_inizio	time	not null

PROIEZIONE		
Attributo	Tipo	Vincoli
id	integer	PK
data	date	not null
id_film	integer	FK(FILM.id)
id_sala	integer	FK(SALA.id)
id_orario	integer	FK(SLOT.id)

Progetto: CineNow	Versione: 1.0
Documento: Database Design Document	Data: 22/11/2024

POSTO_PROIEZIONE		
Attributo	Tipo	Vincoli
id_sala	integer	PKC & FK(SALA.id)
fila	string	PKC & FK(POSTO.fila)
numero_posto	integer	PKC & FK(POSTO.numero)
id_proiezione	integer	PKC & FK(PROIEZIONE.id)
stato	boolean	not null & DEFAULT(false)

PRENOTAZIONE		
Attributo	Tipo	Vincoli
id	integer	PK
email_cliente	string	FK(CLIENTE.email)
id_proiezione	integer	FK(PROIEZIONE.id)

Occupa		
Attributo	Tipo	Vincoli
id_sala	integer	PKC & FK(SALA.id)
fila	string	PKC & FK(POSTO.fila)
numero_posto	integer	PKC & FK(POSTO.numero)
id_proiezione	integer	PKC & FK(PROIEZIONE.id)
id_prenotazione	integer	PKC & FK(PRENOTAZIONE.id)