

Star Cinema - Sistema di gestione di un cinema

Indice generale

Introduzione – Specifiche del Progetto.....	2
Schema E/R.....	3
Entità.....	4
Associazioni.....	6
Schema Relazionale.....	8
Dipendenze Funzionali e Normalizzazione.....	10
SQL - Creazione delle Tabelle.....	12
SQL - Query di Selezione Significative.....	16
SQL - Query di Aggiornamento Significative.....	18
Utenti Significativi.....	19
Interfaccia.....	20

Introduzione – Specifiche del Progetto

Il progetto consiste nella realizzazione di un sito web che simuli la gestione di un vero cinema multisala, con un numero prefissato di sale pari a 4, con uguale numero di posti.

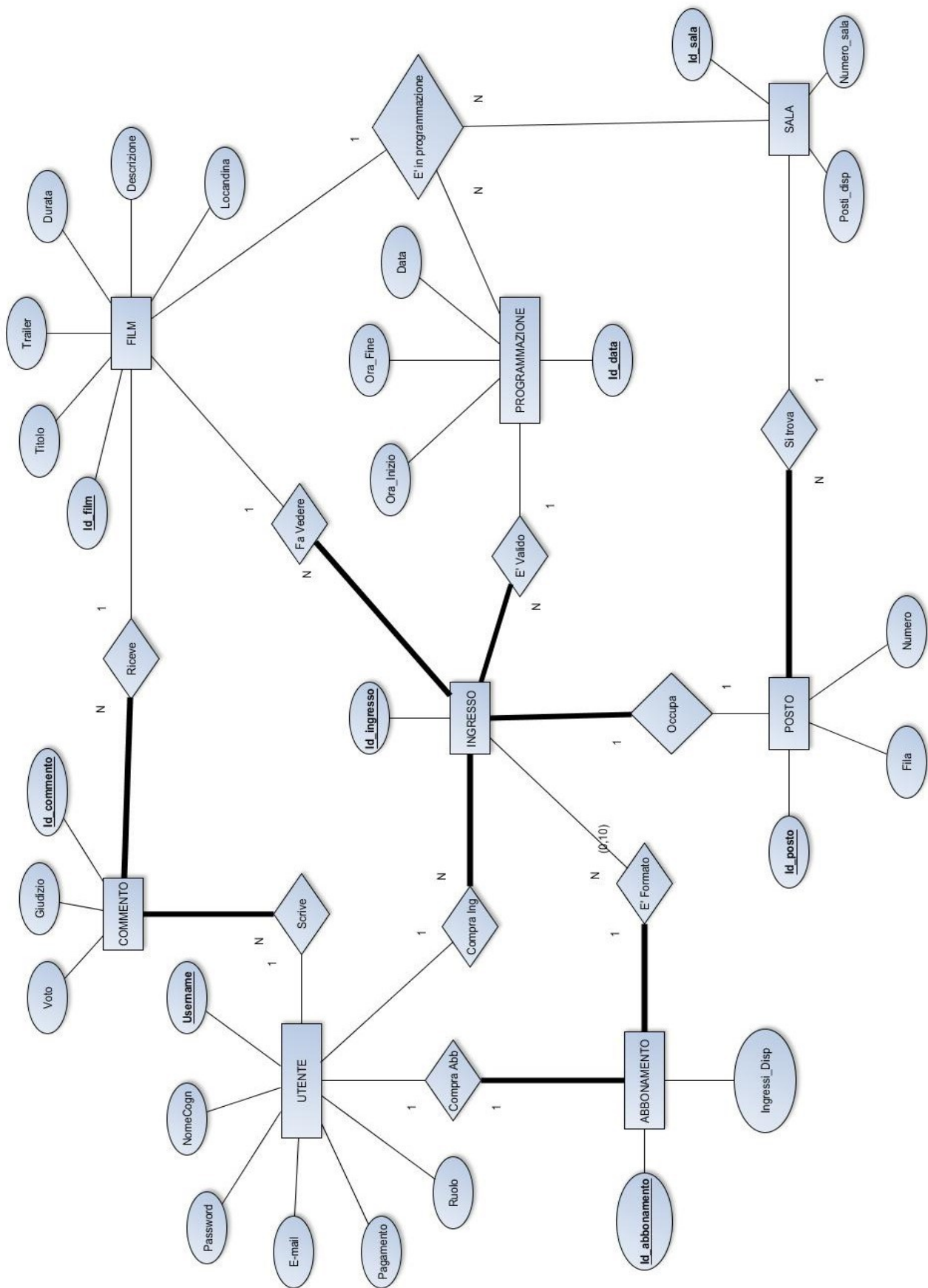
Il sistema deve essere principalmente usato da due tipologie di utenti: Utenti Registrati e Amministratori. Un generico utente che visita il sito, deve essere invogliato a registrarsi, e successivamente effettuare il login, per diventare un Utente Registrato e avere accesso alle funzionalità del sistema. In questo modo, può facilmente vedere la programmazione del sito, i commenti che gli utenti hanno lasciato a un film, acquistare biglietti del cinema selezionando i posti a lui migliori e verificando la disponibilità stessa dei posti. In questo modo si evita che l'utente sia costretto ad andare fisicamente sul posto per poter sapere se vi sono ancora posti o meno per la visione del film.

Un Amministratore, deve avere accesso alle stesse funzionalità del sito, con in più la possibilità di interagirci attivamente tramite l'inserimento di nuovi film, la moderazione dei commenti, l'aggiunta e la modifica della programmazione settimanale e la situazione di ogni sala e in ogni data.

In maniera più specifica, la funzionalità cardine del sito è la facilità di acquisto e gestione dei propri biglietti per i vari spettacoli:

- Dopo aver visionato la programmazione settimanale, visibile anche ad utenti non registrati, l'utente può decidere di acquistare gli ingressi per uno spettacolo solo dopo essersi registrato o dopo aver effettuato il login.
- Scelto il film e la data, si evita che l'utente possa scegliere film non in programmazione o date in cui il film non è presente, viene data la possibilità di scelta dell'orario di visione dello spettacolo con i relativi posti disponibili, ed effettuata la scelta, si ha libero accesso alla selezione dei posti, fino ad un massimo di 5 per utente.
- L'ulteriore utilità della registrazione, è quella di poter modificare in un qualunque momento i biglietti di uno spettacolo non ancora iniziato, in questo modo l'utente non è vincolato alla visione di quel dato spettacolo, ma ha la possibilità di scambiarlo con un altro.

Schema E/R



Entità

Utente

Entità che rappresenta un utente registrato al sito.

I suoi attributi sono:

- Username (chiave primaria): identifica univocamente l'utente. E' deciso dall'utente in fase di registrazione e viene usato nel sito per riconoscere chi lo sta usando.
- NomeCogn: E' il nome e il cognome dell'utente.
- Email: L'email inserita in fase di registrazione dall'utente.
- Password: Il codice che l'utente deve inserire in fase di accesso insieme allo username per essere riconosciuto dal sistema.
- Pagamento: Il codice della carta di credito per effettuare acquisti nel sito.
- Ruolo: Il ruolo che l'utente ha nel sistema. Può essere *Utente* o *Admin*.

Commento

Entità che rappresenta il giudizio inserito da un utente ad un certo film.

I suoi attributi sono:

- Id_commento (chiave primaria): identifica univocamente un commento all'interno del sistema.
- Giudizio: è il commento inserito da un utente riguardo a un film.
- Voto: è il voto da 1 a 5 espresso da un utente riguardo un film.

Film

Entità che rappresenta un film del cinema.

I suoi attributi sono:

- Id_film (chiave primaria): identifica univocamente il film all'interno del sistema.
- Titolo: il nome con cui il film è conosciuto.
- Trailer: l'url per il trailer del film.
- Durata: la durata del film nel formato hh:mm.
- Descrizione: la trama del film.
- Locandina: il nome dell'immagine del film con cui è stata salvata in una delle cartelle del progetto.

Sala

Entità che rappresenta una delle sale in cui i film vengono proiettati.

I suoi attributi sono:

- Id_sala (chiave primaria): identifica univocamente la sala all'interno del sistema.
- Numero_sala: il numero della sala nel cinema.
- Posti_disp: il numero di posti a sedere all'interno della sala.

Posto

Entità che rappresenta un posto a sedere all'interno della sala.

I suoi attributi sono:

- Id_posto (chiave primaria): identifica univocamente il posto all'interno del sistema.
- Fila: E' la lettera corrispondente alla fila all'interno della sala.
- Numero: E' il numero del posto a sedere corrispondente alla fila.

Abbonamento

Entità che rappresenta l'abbonamento di un utente al cinema per acquistare biglietti.

I suoi attributi sono:

- Id_abbonamento (chiave primaria): identifica univocamente l'abbonamento all'interno del sistema.
- Ingressi_disp: il numero di ingressi che un utente può usare per comprare biglietti.

Ingresso

Entità che rappresenta un biglietto del cinema per la visione di un film in programmazione.

I suoi attributi sono:

- Id_ingresso (chiave primaria): identifica univocamente il biglietto nel sistema.

Programmazione

Entità che rappresenta le date e gli orari in cui un film è in programmazione.

I suoi attributi sono:

- Id_data (chiave primaria): identifica univocamente la coppia data-orario all'interno del sistema.
- Data: la data in cui il film è in programmazione.
- Ora_inizio: l'ora in cui inizia il film.
- Ora_fine: l'ora entro cui finisce il film.

Associazioni

Scrive

L'associazione intercorre tra entità di tipo UTENTE e COMMENTO. Identifica la possibilità di un utente di scrivere un commento riguardo a un film. Il COMMENTO ha un vincolo di partecipazione totale in quanto non può esistere un COMMENTO se non è associato ad un UTENTE, mentre un UTENTE può non avere alcun COMMENTO associato.

L'associazione ha cardinalità 1-N in quanto un UTENTE può scrivere un COMMENTO ad ogni film del sistema, ma un COMMENTO può essere scritto e quindi associato solo da un singolo UTENTE.

Riceve

L'associazione intercorre tra entità di tipo COMMENTO e FILM. Identifica la possibilità di un utente di commentare un film. Il COMMENTO ha un vincolo di partecipazione totale in quanto deve essere sempre associato ad un FILM.

L'associazione ha cardinalità 1-N in quanto un FILM può avere un COMMENTO da ogni utente del sistema, ma un COMMENTO può essere associato ad un solo FILM alla volta.

Fa Vedere

L'associazione intercorre tra entità di tipo FILM e INGRESSO. Identifica la possibilità di vedere un film tramite l'acquisto di un biglietto. INGRESSO ha un vincolo di partecipazione totale in quanto non ha senso che esista se non associato ad un FILM.

L'associazione ha cardinalità 1-N in quanto un FILM può essere visto da N utenti che hanno un INGRESSO, ma un INGRESSO dà accesso ad un solo FILM.

E' Valido

L'associazione intercorre tra entità di tipo INGRESSO e PROGRAMMAZIONE. Identifica la data e l'orario in cui il biglietto acquistato è valido per visione di un film. INGRESSO ha un vincolo di partecipazione totale in quanto è sempre associato ad una PROGRAMMAZIONE al momento dell'acquisto.

L'associazione ha cardinalità 1-N in quanto una PROGRAMMAZIONE può avere più INGRESSO che ne permettono l'accesso, ma un INGRESSO è valido solo per una PROGRAMMAZIONE.

Occupi

L'associazione intercorre tra entità di tipo INGRESSO e POSTO. Identifica il posto che un utente ha prenotato al momento dell'acquisto di un biglietto. INGRESSO ha un vincolo di partecipazione totale in quanto deve sempre occupare un POSTO.

L'associazione ha cardinalità 1-1 in quanto a un INGRESSO corrisponde sempre un POSTO, mentre a un POSTO può corrispondere un INGRESSO.

Si Trova

L'associazione intercorre tra entità di tipo POSTO e SALA. Identifica uno dei posti a sedere prenotabili all'interno di una sala del cinema. POSTO ha un vincolo di partecipazione totale poiché non ha senso che esista se non associato alla sala in cui si trova.

L'associazione ha cardinalità 1-N in quanto un POSTO è sempre associato a solo una SALA, ma una SALA ha sempre più di un POSTO.

E' Formato

L'associazione intercorre tra entità di tipo ABBONAMENTO e INGRESSO. Identifica la funzionalità di acquistare un abbonamento per poter comprare uno o più biglietti.

ABBONAMENTO ha un vincolo di partecipazione totale poiché è composto da uno o più entità di tipo INGRESSO.

L'associazione ha cardinalità 1-N con N che può andare da 0 a 10, in quanto ad un ABBONAMENTO può essere associato l'acquisto di più di un INGRESSO, ma un INGRESSO può essere associato ad un solo ABBONAMENTO.

Compra Abb.

L'associazione intercorre tra entità di tipo UTENTE e ABBONAMENTO. Identifica la possibilità dell'acquisto da parte di un utente di un abbonamento del cinema. ABBONAMENTO ha un vincolo di partecipazione totale poiché è sempre associato ad un unico UTENTE.

L'associazione ha cardinalità 1-1 poiché un UTENTE può acquistare massimo un ABBONAMENTO e un ABBONAMENTO è sempre associato ad un solo UTENTE.

Compra Ing.

L'associazione intercorre tra entità di tipo UTENTE e INGRESSO. Identifica la possibilità di un utente di acquistare uno o più ingressi del cinema. INGRESSO ha un vincolo di partecipazione totale in quanto per esistere deve essere sempre collegato all'UTENTE che lo ha acquistato.

L'associazione ha cardinalità 1-N in quanto un UTENTE può acquistare uno o più entità di tipo INGRESSO, ma un INGRESSO è sempre e solo associato all'UTENTE che lo ha acquistato.

E' in programmazione

L'associazione è di tipo ternario e mette in relazione un FILM con la sua PROGRAMMAZIONE e la SALA che occupa. Cioè di volta in volta un film è in programmazione in una certa data e ad un certo orario ed è proiettato in una singola sala.

L'associazione ha cardinalità 1-N-N, poiché un FILM può essere in programmazione in più di una PROGRAMMAZIONE e in visione in più di una SALA.

Schema Relazionale

UTENTI

<u>Username</u>	NomeCogn	Password	E-mail	Pagamento	Ruolo
-----------------	----------	----------	--------	-----------	-------

INGRESSO

<u>Id_ingresso</u>	<u>Id_data</u>	<u>Id_posto</u>	<u>Username</u>	<u>Id_abbonamento</u>	<u>Id_film</u>
--------------------	----------------	-----------------	-----------------	-----------------------	----------------

ABBONAMENTO

<u>Id_abbonamento</u>	Ingressi_Dis	<u>Username</u>
-----------------------	--------------	-----------------

POSTO

<u>Id_posto</u>	Fila	Numero	<u>Id_sala</u>
-----------------	------	--------	----------------

SALA

<u>Id_sala</u>	Posti_disp	Num_sala
----------------	------------	----------

FILM

<u>Id_film</u>	Titolo	Descrizione	Durata	Trailer	Locandina
----------------	--------	-------------	--------	---------	-----------

COMMENTO

<u>Id_commento</u>	Voto	Giudizio	<u>Username</u>	<u>Id_film</u>
--------------------	------	----------	-----------------	----------------

PROGRAMMAZIONE

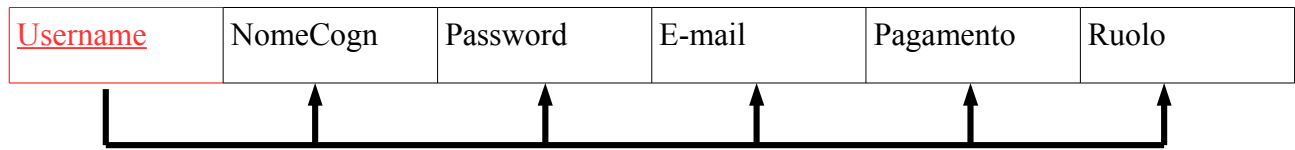
<u>Id_data</u>	Data	Ora_Inizio	Ora_Fine
----------------	------	------------	----------

FILM_SALA_PROGRAMMAZIONE

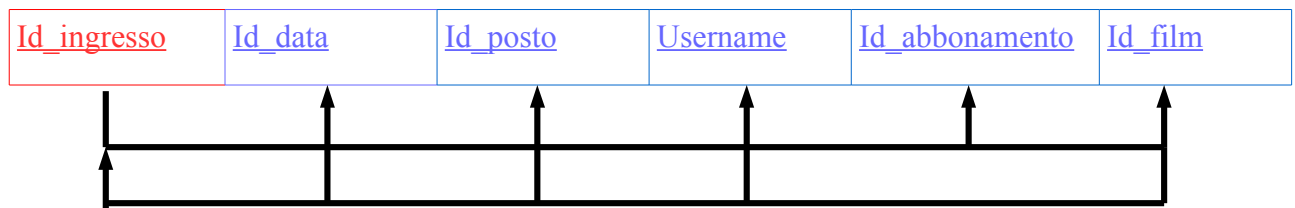
<u>Id_film</u>	<u>Id_data</u>	<u>Id_sala</u>
----------------	----------------	----------------

Dipendenze Funzionali e Normalizzazione

UTENTI

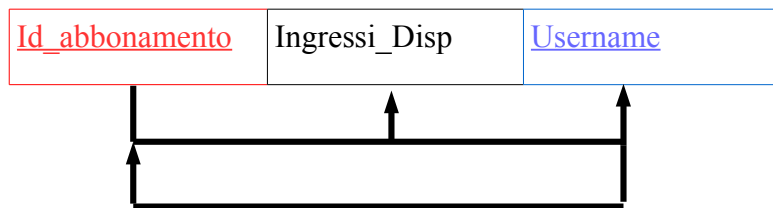


INGRESSO

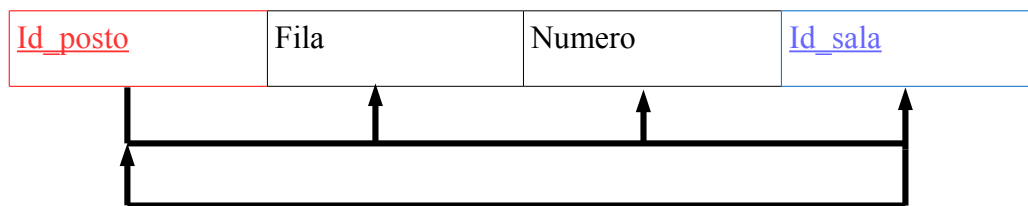


.

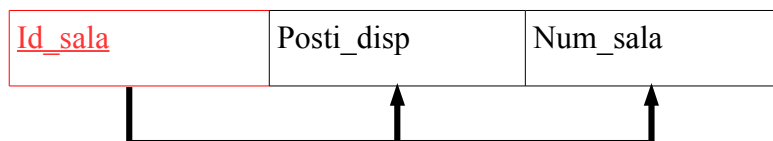
ABBONAMENTO



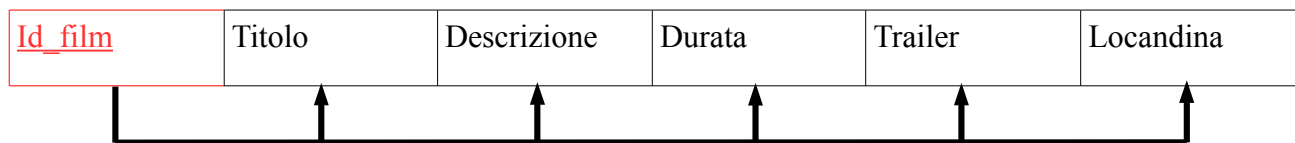
POSTO



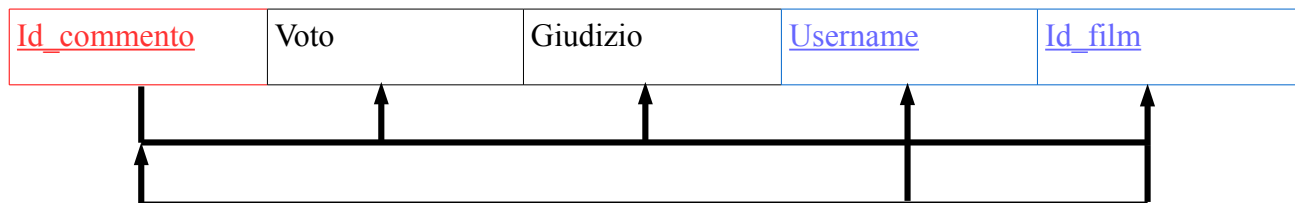
SALA



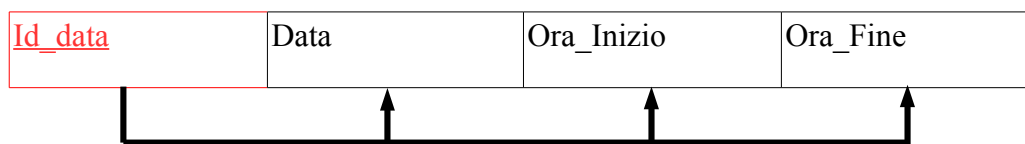
FILM



COMMENTO



PROGRAMMAZIONE



FILM_SALA_PROGRAMMAZIONE



Lo schema relazionale ottenuto dallo schema E/R, risulta essere già in **prima forma normale (1NF)**, in **seconda forma normale (2NF)** e in **terza forma normale (3NF)**.

Infatti tutti gli attributi dello schema E/R sono attributi con valori atomici e il valore di ogni attributo in una tupla è un valore singolo del dominio (**1NF**). Inoltre ogni attributo non-primario di una relazione, dipende funzionalmente in modo completo dalla chiave primaria della relazione stessa (**2NF**). Infine, nessun attributo non-primario di una relazione dipende in modo transitivo dalla chiave primaria (**3NF**).

SQL - Creazione delle Tabelle

Lo schema relazionale considerato per costruire le tabella è quello in **terza forma normale (3NF)**.

```
--
-- Struttura della tabella `abbonamenti`
--

CREATE TABLE IF NOT EXISTS `abbonamenti` (
  `id_abbonamento` int(11) NOT NULL AUTO_INCREMENT,
  `ingressi_disp` int(11) NOT NULL DEFAULT '10',
  `username` varchar(50) NOT NULL,
  PRIMARY KEY (`id_abbonamento`),
  KEY `username` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-----

--
-- Struttura della tabella `commenti`
--

CREATE TABLE IF NOT EXISTS `commenti` (
  `id_commento` int(11) NOT NULL AUTO_INCREMENT,
  `voto` int(11) NOT NULL,
  `giudizio` text NOT NULL,
  `username` varchar(50) NOT NULL,
  `id_film` int(11) NOT NULL,
  PRIMARY KEY (`id_commento`),
  KEY `username` (`username`,`id_film`),
  KEY `id_film` (`id_film`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-----

--
-- Struttura della tabella `film`
--

CREATE TABLE IF NOT EXISTS `film` (
  `id_film` int(11) NOT NULL AUTO_INCREMENT,
  `titolo` text NOT NULL,
  `trailer` text NOT NULL,
  `descrizione` text NOT NULL,
  `durata` time NOT NULL,
  `locandina` text NOT NULL,
  PRIMARY KEY (`id_film`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-----

--
-- Struttura della tabella `film_sala_programmazione`
--
```

```

CREATE TABLE IF NOT EXISTS `film_sala_programmazione` (
  `id_tabella` int(11) NOT NULL AUTO_INCREMENT,
  `id_film` int(11) NOT NULL,
  `id_sala` int(11) NOT NULL,
  `id_data` int(11) NOT NULL,
  PRIMARY KEY (`id_tabella`),
  KEY `id_film` (`id_film`),
  KEY `id_sala` (`id_sala`),
  KEY `id_data` (`id_data`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

```

--
-- Struttura della tabella `ingressi`
--

```

```

CREATE TABLE IF NOT EXISTS `ingressi` (
  `id_ingresso` int(11) NOT NULL AUTO_INCREMENT,
  `id_data` int(11) NOT NULL,
  `id_posto` int(11) NOT NULL,
  `username` varchar(50) NOT NULL,
  `id_abbonamento` int(11) DEFAULT NULL,
  `id_film` int(11) NOT NULL,
  PRIMARY KEY (`id_ingresso`),
  KEY `id_data` (`id_data`),
  KEY `id_posto` (`id_posto`),
  KEY `username` (`username`),
  KEY `id_abbonamento` (`id_abbonamento`),
  KEY `id_film` (`id_film`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

```

--
-- Struttura della tabella `posti`
--

```

```

CREATE TABLE IF NOT EXISTS `posti` (
  `id_posto` int(11) NOT NULL AUTO_INCREMENT,
  `fila` varchar(1) NOT NULL,
  `numero` int(11) NOT NULL,
  `id_sala` int(11) NOT NULL,
  PRIMARY KEY (`id_posto`),
  KEY `id_sala` (`id_sala`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

```

--
-- Struttura della tabella `programmazione`
--

```

```

CREATE TABLE IF NOT EXISTS `programmazione` (
  `id_data` int(11) NOT NULL AUTO_INCREMENT,
  `data` date NOT NULL,
  `ora_inizio` time NOT NULL,

```

```

    `ora_fine` time NOT NULL,
    PRIMARY KEY (`id_data`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-----

--
-- Struttura della tabella `sale`
--

CREATE TABLE IF NOT EXISTS `sale` (
  `id_sala` int(11) NOT NULL AUTO_INCREMENT,
  `posti_disp` int(11) NOT NULL DEFAULT '200',
  `numero_sala` int(11) NOT NULL,
  PRIMARY KEY (`id_sala`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-----

--
-- Struttura della tabella `utenti`
--

CREATE TABLE IF NOT EXISTS `utenti` (
  `username` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  `name` varchar(50) NOT NULL,
  `surname` varchar(50) NOT NULL,
  `email` varchar(100) NOT NULL,
  `credit_card` int(11) DEFAULT NULL,
  `role` varchar(10) NOT NULL DEFAULT 'user',
  `active` tinyint(1) NOT NULL DEFAULT '0' COMMENT 'per conferma nuovo
utente',
  PRIMARY KEY (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Limiti per la tabella `abbonamenti`
--

ALTER TABLE `abbonamenti`
  ADD CONSTRAINT `abbonamenti_ibfk_1` FOREIGN KEY (`username`) REFERENCES
`utenti` (`username`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Limiti per la tabella `commenti`
--

ALTER TABLE `commenti`
  ADD CONSTRAINT `commenti_ibfk_1` FOREIGN KEY (`username`) REFERENCES
`utenti` (`username`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `commenti_ibfk_2` FOREIGN KEY (`id_film`) REFERENCES `film`
(`id_film`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Limiti per la tabella `film_sala_programmazione`
--

ALTER TABLE `film_sala_programmazione`
  ADD CONSTRAINT `film_sala_programmazione_ibfk_1` FOREIGN KEY (`id_film`)
REFERENCES `film` (`id_film`) ON DELETE CASCADE ON UPDATE CASCADE,

```

```

    ADD CONSTRAINT `film_sala_programmazione_ibfk_2` FOREIGN KEY (`id_sala`)
REFERENCES `sale` (`id_sala`) ON DELETE CASCADE ON UPDATE CASCADE,
    ADD CONSTRAINT `film_sala_programmazione_ibfk_3` FOREIGN KEY (`id_data`)
REFERENCES `programmazione` (`id_data`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Limiti per la tabella `ingressi`
--
ALTER TABLE `ingressi`
    ADD CONSTRAINT `ingressi_ibfk_1` FOREIGN KEY (`username`) REFERENCES
`utenti` (`username`) ON DELETE CASCADE ON UPDATE CASCADE,
    ADD CONSTRAINT `ingressi_ibfk_2` FOREIGN KEY (`id_abbonamento`) REFERENCES
`abbonamenti` (`id_abbonamento`) ON DELETE CASCADE ON UPDATE CASCADE,
    ADD CONSTRAINT `ingressi_ibfk_3` FOREIGN KEY (`id_posto`) REFERENCES
`posti` (`id_posto`) ON DELETE CASCADE ON UPDATE CASCADE,
    ADD CONSTRAINT `ingressi_ibfk_4` FOREIGN KEY (`id_film`) REFERENCES `film`
(`id_film`) ON DELETE CASCADE ON UPDATE CASCADE,
    ADD CONSTRAINT `ingressi_ibfk_5` FOREIGN KEY (`id_data`) REFERENCES
`programmazione` (`id_data`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Limiti per la tabella `posti`
--
ALTER TABLE `posti`
    ADD CONSTRAINT `posti_ibfk_1` FOREIGN KEY (`id_sala`) REFERENCES `sale`
(`id_sala`) ON DELETE CASCADE ON UPDATE CASCADE;

```

SQL - Query di Selezione Significative

Vengono riportate alcune query di selezione significative per il corretto funzionamento del sistema. Le query sono tutte inviate dal server tramite API di Java.

Recupero dei posti occupati per uno spettacolo

```
SELECT *  
FROM `ingressi` AS I JOIN `posti` AS P ON I.id_posto=P.id_posto  
WHERE I.id_film=id_film AND P.id_sala=id_sala AND I.id_data=id_data
```

- *id_film* permette di identificare univocamente il film
- *id_sala* permette di identificare univocamente la sala
- *id_data* permette di identificare univocamente la data

Vengono recuperati tutti i posti occupati per un certo spettacolo. Viene usata nell'acquisto e nella modifica dei biglietti per evidenziare quali posti sono già prenotati e quali no.

Ricerca film tramite data di programmazione

```
SELECT DISTINCT F.id_film, F.titolo  
FROM `film` AS F JOIN `film_sala_programmazione` AS FSP ON  
F.id_film=FSP.id_film  
JOIN `programmazione` AS P ON P.id_data=FSP.id_data  
WHERE P.data='data'  
ORDER BY F.titolo
```

- *data* corrisponde alla data di programmazione in cui si vuole vedere la lista dei film. E' nel formato yyyy-MM-dd

Vengono recuperati tutti i film che sono in programmazione in quella data. Tramite **DISTINCT** si evita di inserire doppioni, ed inoltre il risultato è ordinato in base al titolo.

Recupero coppie data-sala in cui un film è in programmazione in un periodo dato da due date

```
SELECT *  
FROM `film_sala_programmazione` AS FSP JOIN `programmazione` AS P ON  
FSP.id_data=P.id_data JOIN `sale` AS S ON FSP.id_sala=S.id_sala  
WHERE FSP.id_film=id_film AND P.data BETWEEN from AND to  
ORDER BY P.data, S.numero_sala, P.ora_inizio
```

- *id_film* serve per identificare il film
- *from* è la data di inizio del periodo che si considera
- *to* è la data finale del periodo che si considera

Per un certo film, vengono recuperate tutte le sale e le rispettive date e orari in cui è in programmazione in un certo lasso di tempo. La query è usata per la creazione della programmazione settimanale del cinema.

Recupero lista ingressi associati ad un utente

```
SELECT *  
FROM `ingressi` AS I JOIN `programmazione` AS P ON I.id_data=P.id_data  
WHERE I.username=username  
ORDER BY P.data DESC, P.ora_inizio
```

- *username* serve per identificare l'utente

Vengono recuperati tutti gli ingressi associati ad un utente, che successivamente vengono inseriti in una lista per essere elaborati dal server. Sono ordinati per data, dal più recente in poi, e per ora_inizio.

SQL - Query di Aggiornamento Significative

Vengono riportate alcune query di aggiornamento ed eliminazione significative per il corretto funzionamento del sistema. Le query sono tutte inviate dal server tramite API di Java.

Aggiornamento di un film presente nel database

```
UPDATE `film`  
SET  
`titolo`=titolo, `trailer`=trailer, `descrizione`=descrizione, `durata`=durata,  
`locandina`=locandina  
WHERE `id_film`=id_film
```

- *titolo* corrisponde al nuovo titolo che si vuole assegnare al film
- *trailer* corrisponde al nuovo url per vedere il trailer di un film
- *descrizione* corrisponde alla nuova descrizione del film
- *durata* corrisponde alla durata del film nel formato hh:mm
- *locandina* corrisponde all'url dell'immagine ufficiale del film
- *id_film* identifica il film che si vuole modificare

Viene utilizzata per l'aggiornamento di un film già presente nel database.

Aggiornamento dati di un utente nel database

```
UPDATE `utenti`  
SET `name`=name, `surname`=surname, `email`=email, `credit_card`=creditcard  
WHERE `username`=username
```

- *name* corrisponde al nuovo nome che si vuole dare all'utente
- *surname* corrisponde al nuovo cognome che si vuole dare all'utente
- *email* corrisponde alla nuova email che l'utente vuole usare
- *creditcard* corrisponde al codice della carta di credito
- *username* permette di identificare univocamente l'utente da aggiornare

Viene usata da un utente quando vuole aggiornare i suoi dati dalla pagina utente. Gli unici dati che non possono essere modificati sono lo username e la password.

Utenti Significativi

UTENTE REGISTRATO

Un utente registrato ha la possibilità di sfruttare appieno le funzionalità principali del sito, quali la visione della programmazione con l'acquisto di uno o più ingressi per uno spettacolo, oppure la possibilità di commentare e/o visualizzare i commenti di altri utenti ad un film.

AMMINISTRATORE

Un amministratore ha il pieno controllo del sito. Può fare tutto ciò che un utente registrato è in grado di fare, ma allo stesso tempo gestisce il sito. Può aggiungere, modificare o eliminare un film, inserirlo in programmazione, controllare la situazione di ogni sala del cinema o del cinema in ogni data della settimana. Puoi inoltre moderare i commenti degli altri utenti, eliminandoli e inviando una mail di notifica all'utente interessato.

Star(k) Cinema Home

Acquista Biglietto per un Film...

Seleziona una data...

Conferma

Questa Settimana



1 2

Programmazione Settimanale Star Cinema

2014-09-14

2014-09-15

2014-09-16

2014-09-17

2014-09-18

2014-09-19

2014-09-20

Film	Orari	Sala - Posti Disponibili
Iron Man 3	16:00-18:40 <-> 20:00-22:40 <->	Sala 2 - 200 Sala 2 - 200
Sin City - Una donna per cui uccidere	15:00-17:13 <-> 17:30-19:43 <->	Sala 3 - 200 Sala 3 - 200
Guardiani della Galassia	16:00-18:31 <-> 19:00-21:31 <->	Sala 1 - 200 Sala 1 - 200
Hancock	17:00-19:02 <-> 20:00-22:02 <->	Sala 4 - 200 Sala 4 - 200
Il lato positivo - Silver Linings Playbook	20:00-22:32 <->	Sala 3 - 200

Elimina	Titolo	Azioni disponibili	
<input type="checkbox"/>	Forrest Gump	Modifica	Elimina
<input type="checkbox"/>	Guardiani della Galassia	Modifica	Elimina
<input type="checkbox"/>	Hancock	Modifica	Elimina
<input type="checkbox"/>	Il lato positivo - Silver Linings Playbook	Modifica	Elimina
<input type="checkbox"/>	Io, Robot	Modifica	Elimina
<input type="checkbox"/>	Iron Man 3	Modifica	Elimina
<input type="checkbox"/>	Parto col folle	Modifica	Elimina
<input type="checkbox"/>	Sex Tape - Finiti in rete	Modifica	Elimina
<input type="checkbox"/>	Sin City - Una donna per cui uccidere	Modifica	Elimina

Seleziona Posti a Sedere

Fila	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
C	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
D	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
E	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
F	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
G	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
H	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
J	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

☐ Seleziona per usare abbonamento - 10 ingressi disponibili

Conferma

Indietro