

Traduzione inversa - Sistema di gestione Cinema

Dato il seguente codice, **disegna un ER**:

```
1 CREATE TABLE Cinema (  
2     CinemaID INT PRIMARY KEY,  
3     Name VARCHAR(100) NOT NULL,  
4     Address VARCHAR(255) NOT NULL,  
5     Phone VARCHAR(20)  
6 );  
7  
8 CREATE TABLE Theater (  
9     TheaterID INT PRIMARY KEY,  
10    CinemaID INT,  
11    Name VARCHAR(50) NOT NULL,  
12    Capacity INT NOT NULL,  
13    ScreenType VARCHAR(50),  
14    FOREIGN KEY (CinemaID) REFERENCES Cinema(CinemaID)  
15 );  
16  
17 CREATE TABLE Movie (  
18     MovieID INT PRIMARY KEY,  
19     Title VARCHAR(255) NOT NULL,  
20     Director VARCHAR(100),  
21     ReleaseDate DATE,  
22     DurationMinutes INT,  
23     Rating VARCHAR(5)  
24 );  
25  
26 CREATE TABLE Showtime (  
27     ShowtimeID INT PRIMARY KEY,  
28     MovieID INT,  
29     TheaterID INT,  
30     ShowDateTime DATETIME NOT NULL,  
31     Price DECIMAL(5,2) NOT NULL,  
32     FOREIGN KEY (MovieID) REFERENCES Movie(MovieID),  
33     FOREIGN KEY (TheaterID) REFERENCES Theater(TheaterID)  
34 );  
35  
36 CREATE TABLE Ticket (  
37     TicketID INT PRIMARY KEY,  
38     ShowtimeID INT,  
39     SeatNumber VARCHAR(10) NOT NULL,  
40     PurchasedDateTime DATETIME NOT NULL,  
41     CustomerID INT,  
42     FOREIGN KEY (ShowtimeID) REFERENCES Showtime(ShowtimeID),  
43     FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)  
44 );  
45  
46 CREATE TABLE Customer (  
47     CustomerID INT PRIMARY KEY,  
48     FirstName VARCHAR(50) NOT NULL,
```

```

49     LastName VARCHAR(50) NOT NULL,
50     Email VARCHAR(100),
51     PhoneNumber VARCHAR(20)
52 );
53
54 CREATE TABLE Review (
55     ReviewID INT PRIMARY KEY,
56     MovieID INT,
57     CustomerID INT,
58     ReviewText TEXT,
59     Rating INT CHECK (Rating >= 1 AND Rating <= 5),
60     ReviewDate DATETIME NOT NULL,
61     FOREIGN KEY (MovieID) REFERENCES Movie(MovieID),
62     FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
63 );
64
65 CREATE TABLE Employee (
66     EmployeeID INT PRIMARY KEY,
67     CinemaID INT,
68     FirstName VARCHAR(50) NOT NULL,
69     LastName VARCHAR(50) NOT NULL,
70     Position VARCHAR(50),
71     HireDate DATE,
72     FOREIGN KEY (CinemaID) REFERENCES Cinema(CinemaID)
73 );

```

Richieste di Creazione di Viste

1. Vista dei Film in Programmazione

Creare una vista `FilmsInProgrammation` che mostri i titoli dei film, la data di inizio programmazione, la durata e la classificazione per età. Questa vista aiuterà il personale e i clienti a vedere rapidamente quali film sono attualmente in programmazione.

2. Vista Posti Disponibili per Spettacolo

Creare una vista `AvailableSeatsForShow` che, per ogni spettacolo, mostri il numero totale di posti nella sala e quanti sono ancora disponibili. Questa vista è essenziale per il personale alla biglietteria per gestire le vendite dei biglietti.

3. Vista Incassi Totali per Film

Generare una vista `TotalEarningsPerMovie` che elenchi ogni film insieme agli incassi totali generati. Questa informazione è cruciale per la direzione per valutare il successo commerciale dei film.

4. Vista Recensioni Recenti

Creare una vista `RecentReviews` che mostri le ultime recensioni lasciate dai clienti, includendo il titolo del film, la valutazione, il testo della recensione e la data. Questo permetterà al personale e alla direzione di monitorare il feedback dei clienti in tempo reale.

Richieste di Creazione di Stored Procedures

1. Procedura di Acquisto Biglietti

Creare una stored procedure `PurchaseTicket` che permetta di acquistare un biglietto per uno spettacolo, specificando l'ID dello spettacolo, il numero del posto e l'ID del cliente. La procedura dovrebbe verificare la disponibilità del posto e registrare l'acquisto.

2. Procedura di Aggiornamento Programmazione Film

Implementare una stored procedure `UpdateMovieSchedule` che permetta di aggiornare gli orari degli spettacoli per un determinato film. Questo include la possibilità di aggiungere o rimuovere spettacoli dall'agenda.

3. Procedura di Inserimento Nuovo Film

Sviluppare una stored procedure `InsertNewMovie` che consenta di inserire un nuovo film nel sistema, richiedendo tutti i dettagli pertinenti come titolo, regista, data di uscita, durata e classificazione.

4. Procedura per Registrare una Recensione

Creare una stored procedure `SubmitReview` che consenta ai clienti di lasciare una recensione per un film, comprensiva di valutazione, testo e data. Questa procedura dovrebbe verificare che il cliente abbia effettivamente acquistato un biglietto per il film in questione prima di permettere la pubblicazione della recensione.