

Database Design Documents

Subject: Database for Movie System

Group Member: Xingyao Wu, Jiayuan Chen, Li Cai

Description of the business problem:

Nowadays, young generations have changed their way of watching movies in the cinemas. Traditionally, people were more used to walk into the cinema and look up their favorite movie through the screening schedule board. However, this way had turned into a problem that people couldn't know anything about the movies' ticket availability and their ratings before walking in. To solve this problem, more and more cinema companies and movie service providers have introduced their own web or mobile applications. With the applications, users can know the movie schedule just in time, and make it possible to lock in and pay for their movie tickets earlier. To support these applications, a well-designed relational database should be implemented.

Design Decision:

1. Person:

A. **Why included:** This is the generic entity which represents a natural person. It provides rather detailed information about a user or an employee.

B. Attributes:

- a) Person_ID
- b) SSN
- c) FirstName
- d) LastName
- e) Gender
- f) Birthday

2. User:

A. **Why included:** Every person can register as a user of this database, and then many functions can initially be used. Introducing a User entity can not only provide the details

of this user, but also makes sure that a single person is only registered as a single user.

B. **How related:** The user entity has a one-to-one mapping to the Person.

C. **Attributes:**

- a) User_ID
- b) Email
- c) Password
- d) Username
- e) Person_ID

3. **User's Watchlist:**

A. **Why included:** An important function of this database is to enable a personal watchlist for every user. Users may freely add or delete any movie into his or her own watchlist, who helps the user to arrange his or her own movie watching plan.

B. **How related:** The User's Watchlist entity has a one-to-many mapping to the Movie and the User. It is connected with a user and movie by importing a foreign key of its id.

C. **Attributes:**

- a) Watchlist's_ID
- b) User's_ID
- c) Movie's_ID

4. **Employee:**

A. **Why included:** Employees of theatres need to be recorded for further functions, such as booking a ticket for a user, etc. Also, this entity along with User and Person entity together, makes sure a person can be a user and an employee at the same time.

B. **How related:** The employee entity has a one-to-one mapping to the Person.

C. **Attributes:**

- a) Employee_ID
- b) Description
- c) Theatre_ID

d) Person_ID

5. **Movie Distributor:**

A. **Why included:** By collecting the information about a movie's distributor, this database helps users to make a better decision when booking a ticket.

B. **How related:** The movie distributor entity has a one-to-many mapping to the Movie.

C. **Attributes:**

- a) Distributor_ID
- b) Distributor_Name
- c) Distributor_Description
- d) Distributor_Company

6. **Theatre:**

A. **Why included:** A theatre is not only where movies take place, but also a catalogue for users to browse a list of movies.

B. **How related:** The theatre entity has a one to many mapping to the Employee and Showroom.

C. **Attributes:**

- a) Theatre_ID
- b) Theatre_Name
- c) Description

7. **Theatre Address:**

A. **Why included:** There were some discussion around this entity. We finally agreed on introducing it to avoid too long an attribute existing in the Theatre entity.

B. **How related:** The treater address entity which has a one-to-one mapping with the Theatre.

C. **Attributes:**

- a) Theatre_Address_ID
- b) City
- c) State
- d) Zipcode
- e) Address_Line_1
- f) Address_Line_2

8. **Showroom:**

- A. **Why included:** To keep the movie playing well arranged, each theatre needs a list of showrooms to do the scheduling. Users also needs to see which showroom is his or her movie is screening in after booking a ticket.
- B. **How related:** The showroom entity which has a one-to-many mapping with ticket and screening.
- C. **Attributes:**
 - a) Showroom_ID
 - b) Showroom_Name
 - c) Theatre_ID

9. **Ticket:**

- A. **Why included:** One of the key functions of this database is to let the users book tickets. With this entity included, the database can not only record ticket selling information for users, but also provide data source for theatres to perform data analysis and data mining, to decide which movie is the best seller.
- B. **How related:** The ticket entity has a many-to-one mapping with screening schedule. As to users, he or she can have many or no tickets.
- C. **Attributes:**
 - a) Ticket_ID
 - b) Ticket_Price
 - c) Screening_ID
 - d) User_ID

10. **Screening Schedule:**

- A. **Why included:** Screening schedule is a critical parameter to be considered when a user booking tickets.
- B. **How related:** The screening schedule entity has a many-to-one mapping with Movie, Showroom and ticket.
- C. **Attributes:**
 - a) Screening_ID
 - b) Showroom_ID
 - c) Movie_ID

d) Showtime

11. **Movie:**

- A. **Why included:** Movie is one of the core entities in this database. It provides the details of a movie for users, helps them decide whether to book a ticket or not.
- B. **How related:** The movie entity which has a one-to-many mapping with Screening Schedule, Movie Genre, Movie Actor Catalog, Movie Director Catalog, Movie Rating, Award and Movie News Catalog.
- C. **Attributes:**
 - a) Movie_ID
 - b) Movie_Title
 - c) Movie_Duration
 - d) Movie_Release_Date
 - e) Movie_Country
 - f) Movie_Description
 - g) Distributor_ID
 - h) AverageRating

12. **Box Office:**

- A. **Why included:** This entity records information interested by both users and theatres. The box office income of a movie provide perfect data source for theatres to do analysis, while providing auxiliary information for users to book a ticket.
- B. **How related:** By sharing a common movie ID attribute with a movie, a single movie has a certain box office.
- C. **Attributes:**
 - a) Movie_ID
 - b) Movie_Income
 - c) Movie_Deadline

13. **Movie Genre Catalogue:**

- A. **Why included:** This is introduced to normalize the many-to-many relationship between movies and its genres.
- B. **How related:** Two primary key of the two entities to be related are associated here.

C. **Attributes:**

- d) Genre_ID
- e) Movie_ID

14. **Genre:**

A. **Why Included:** This genre included the genre information used for users to select their favorite movie.

B. **How related:** Genre entity has a many-to-one mapping with MovieGenreCatalog.

C. **Attributes:**

- a) Genre_ID
- b) Genre_Name
- c) Genre_Desicription

15. **Trailer:**

A. **Why included:** It is convenient for users to see a trailer of a movie before buying its ticket.

B. **How related:** A movie can have zero or more trailers, but a trailer can only be about one movie.

C. **Attributes:**

- a) Trailer ID
- b) Trailer Duration
- c) Trailer Description

16. **Movie News Catalog:**

A. **Why included:** This is introduced to normalize the many-to-many relationship between movies and news.

B. **How related:** Two primary key of the two entities to be related are associated here.

C. **Attributes:**

- a) News_ID
- b) Movie ID

17. **News:**

A. **Why included:** For users, news of a movie can be critical when making a decision. This entity stores information of all news about movie for users to look up.

B. **How related:** The news entity has a one-to-many mapping with movie news catalog.

C. **Attributes:**

- a) News_ID
- b) News_Title
- c) News_Description
- d) News_Source
- e) News_Editor

18. Award:

A. **Why included:** It is also important for users to see the awards won by a movie.

B. **How related:** A movie can win zero or more awards, but an award can only be present to one movie.

C. **Attributes:**

- a) Award_ID
- b) Award_Title
- c) Award_Description
- d) Award_Year
- e) Movie_ID

19. Movie Rating: The movie rating entity has a one to one mapping with movie.

A. **Why included:** Movie rating is another important function of this database. Users can rate a movie according to their own favor, and the average of history ratings that the movie received can be displayed for all users.

B. **How related:** A movie can have zero or more rating, but a rating can only be about one movie.

C. **Attributes:**

- a) Rating_ID
- b) Rating
- c) Rating_Website
- d) Movie_ID

20. Movie Director Catalog:

A. **Why included:** This is introduced to normalize the many-to-many relationship between movies and directors.

B. **How related:** Two primary key of the two entities to be related are associated here.

C. **Attributes:**

- a) Movie_ID
- b) Director_ID

21. Director:

A. **Why included:** Many users may consider the director of a movie makes a weighted factor in their purchasing behavior. It is also important to make it accessible for users.

B. **How related:** The director entity has a one-to-many mapping with MovieDirectorCatalog. And the director entity has one-to-one mapping with person entity.

C. **Attribute:**

- a) Director_ID
- b) Director_Description
- c) Person_ID

22. Movie Actor Catalog:

A. **Why included:** This is introduced to normalize the many-to-many relationship between movies and actors.

B. **How related:** Two primary key of the two entities to be related are associated here.

C. **Attributes:**

- a) Movie_ID
- b) Actor_ID

23. Actor:

A. **Why included:** Many users may consider the director of a movie makes a weighted factor in their purchasing behavior. It is also important to make it accessible for users.

B. **How related:** The actor entity has a one-to-many mapping with MovieActorCatalog. And the Actor entity has one-to-one mapping with person entity.

C. **Attributes:**

- a) Actor_ID

- b) Actor_Description
- c) Person_ID