

DBMS Task 2

MOVIE TV SHOWS INFORMATION MANAGEMENT

DESCRIPTION

The TV and movie show database is designed to store information about movies, TV shows, actors, directors, genres, and user ratings. Users can browse, search, and rate their favorite movies and TV shows.

UNNORMALIZED TABLES

movies:

- MovieID (Primary Key)
- Title
- ReleaseYear
- Duration
- Description
- Director
- Genre

tvshows:

- ShowID (Primary Key)
- Title
- StartYear
- EndYear
- Description
- Creator
- Genre

actors:

- ActorID (Primary Key)
- Name
- BirthDate
- Gender
- Country

moviecast:

- MovieID (Foreign Key Referencing movies: MovieID)
- ActorID (Foreign Key Referencing actors: ActorID)
- RoleName

showcast:

- ShowID (Foreign Key Referencing tvshows: ShowID)
- ActorID (Foreign Key Referencing actors: ActorID)
- RoleName

movieratings:

- RatingID (Primary Key)
- MovieID (Foreign Key referencing movies: MovieID)
- UserID (Foreign Key referencing users: UserID)
- RatingValue
- ReviewText
- TimeStamp

showratings:

- RatingID (Primary Key)
- ShowID (Foreign Key referencing tvshows: ShowID)
- UserID (Foreign Key referencing users: UserID)
- RatingValue
- ReviewText

- TimeStamp

users:

- UserID (Primary Key)
- Username
- Email
- Password
- UserType

1NF NORMALIZED TABLES

Here, in the **movies** table the name of **Director** is converted to its atomic form as **DirectorFirstName** and **DirectorLastName**. Similarly for **Creator** attribute of **tvshows** table.

movies:

- MovieID (Primary Key)
- Title
- ReleaseYear
- Duration
- Description
- DirectorFirstName
- DirectorLastName
- Genre

tvshows:

- ShowID (Primary Key)
- Title
- StartYear
- EndYear
- Description
- CreatorFirstName

- CreatorLastName
- Genre

actors:

- ActorID (Primary Key)
- FirstName
- LastName
- BirthDate
- Gender
- Country

moviecast:

- MovieID (Foreign Key Referencing movies: MovieID)
- ActorID (Foreign Key Referencing actors: ActorID)
- RoleName

showcast:

- ShowID (Foreign Key Referencing tvshows: ShowID)
- ActorID (Foreign Key Referencing actors: ActorID)
- RoleName

movieratings:

- RatingID (Primary Key)
- MovieID (Foreign Key referencing movies: MovieID)
- UserID (Foreign Key referencing users: UserID)
- RatingValue
- ReviewText
- TimeStamp

showratings:

- RatingID (Primary Key)
- ShowID (Foreign Key referencing tvshows: ShowID)
- UserID (Foreign Key referencing users: UserID)
- RatingValue
- ReviewText
- TimeStamp

users:

- UserID (Primary Key)
- Username
- Email
- Password
- UserType

2NF NORMALIZED TABLES

Here, in the **movies** table, the **DirectorFirstName** and **DirectorLastName** are removed, and they are added to a new table **directors** to remove the partial dependencies in the table. Similar in the case of **tvshows** table too.

movies:

- MovieID (Primary Key)
- Title
- ReleaseYear
- Duration
- Description
- DirectorID (Foreign Key References Director: DirectorID)
- Genre

tvshows:

- ShowID (Primary Key)

- Title
- StartYear
- EndYear
- Description
- CreatorID (Foreign Key References Director: DirectorID)
- Genre

directors:

- DirectorID (Primary Key)
- FirstName
- LastName

actors:

- ActorID (Primary Key)
- FirstName
- LastName
- BirthDate
- Gender
- Country

moviecast:

- MovieID (Foreign Key Referencing movies: MovieID)
- ActorID (Foreign Key Referencing actors: ActorID)
- RoleName

showcast:

- ShowID (Foreign Key Referencing tvshows: ShowID)
- ActorID (Foreign Key Referencing actors: ActorID)
- RoleName

movieratings:

- RatingID (Primary Key)
- MovieID (Foreign Key referencing movies: MovieID)
- UserID (Foreign Key referencing users: UserID)
- RatingValue
- ReviewText
- TimeStamp

showratings:

- RatingID (Primary Key)
- ShowID (Foreign Key referencing tvshows: ShowID)
- UserID (Foreign Key referencing users: UserID)
- RatingValue
- ReviewText
- TimeStamp

users:

- UserID (Primary Key)
- Username
- Email
- Password
- UserType

3NF NORMALIZED TABLES

Here, Genre attribute in **movies** and **tvshows** table is added to another table **genre** and the respective genres of each movie or tv show is mapped to the corresponding movie or tv show using two tables **moviegenres** and **showgenres**.

movies:

- MovieID (Primary Key)
- Title
- ReleaseYear
- Duration
- Description
- DirectorID (Foreign Key References Director: DirectorID)

tvshows:

- ShowID (Primary Key)
- Title
- StartYear
- EndYear
- Description
- CreatorID (Foreign Key References Director: DirectorID)

directors:

- DirectorID (Primary Key)
- FirstName
- LastName

actors:

- ActorID (Primary Key)
- FirstName
- LastName
- BirthDate
- Gender
- Country

genres:

- GenreID (Primary Key)

- GenreName

moviegenres:

- MovieID (Foreign Key referencing movies)
- GenreID (Foreign Key referencing genres)

showgenres:

- ShowID (Foreign Key referencing tvshows)
- GenreID (Foreign Key referencing genres)

roles:

- RoleID (Primary Key)
- RoleName

moviecast:

- MovieID (Foreign Key Referencing movies: MovieID)
- ActorID (Foreign Key Referencing actors: ActorID)
- RoleID (Foreign Key Referencing roles: roleID)

showcast:

- ShowID (Foreign Key Referencing tvshows: ShowID)
- ActorID (Foreign Key Referencing actors: ActorID)
- RoleName

movieratings:

- RatingID (Primary Key)
- MovieID (Foreign Key referencing movies: MovieID)
- UserID (Foreign Key referencing users: UserID)
- RatingValue
- ReviewText

- TimeStamp

showratings:

- RatingID (Primary Key)
- ShowID (Foreign Key referencing tvshows: ShowID)
- UserID (Foreign Key referencing users: UserID)
- RatingValue
- ReviewText
- TimeStamp

users:

- UserID (Primary Key)
- Username
- Email
- Password
- UserType

DETAILED INFORMATION

- **movies and tvshows:** Store basic information about movies and TV shows, such as title, release/start year, duration, and description.
- **directors and actors:** Capture details about directors and actors involved in the productions.
- **genres:** Maintain a list of genres.
- **moviegenres and showgenres:** Establish many-to-many relationships between movies and genres, and TV shows and genres.
- **roles and cast (moviecast and showcast):** Define various roles (e.g., lead actor, supporting actor) and link actors to roles in both movies and TV shows.
- **Ratings (movieratings and showratings):** Store user ratings, reviews, and timestamps for both movies and TV shows.

- **users:** Keep track of user information, including username, email, password (hashed), and user type.

ENTITY RELATIONSHIP DIAGRAM

