Relational Model

GROUP NO 07

To design a relational model for a music database system with the given entities, we can create tables for each entity, specifying their attributes and relationships. Here's a simplified example:

1. User
- UserID (Primary Key)
- Username
- SubscriptionID (Foreign Key)

2. Playlist
- PlaylistID (Primary Key)
- UserID (Foreign Key)
- PlaylistName

3. Song
- SongID (Primary Key)
- Title
- Duration
- ComposerID (Foreign Key)
- SingerID (Foreign Key)
- Artists
- Albums (with AlbumID, Title, Release Date)
- Songs (with SongID, Title, Duration)
- lyrics

4. Recommendation
- RecommendationID (Primary Key)
- UserID (Foreign Key)
- SongID (Foreign Key)

5. Movie
- MovieID (Primary Key)
- Title
- ReleaseDate

6. Platform
- PlatformID (Primary Key)
- PlatformName

7. SongOnPlatform
- SongOnPlatformID (Primary Key)
- SongID (Foreign Key)
- PlatformID (Foreign Key)
- ReleaseDate

8. Creator
- CreatorID (Primary Key)
- Name

9. CreatorSong
- CreatorSongID (Primary Key)
- CreatorID (Foreign Key)
- SongID (Foreign Key)

10. Singer
- SingerID (Primary Key)
- Name

11. Composer
- ComposerID (Primary Key)
- Name

12. Subscription
- SubscriptionID (Primary Key)
- SubscriptionType
- Price

MULTIVALUED ATTRIBUTES
1. ARTISTname
2. PlayList

//# WEAK ENTITY
1. //Playlist
2. //Subscription
3. //SongOnPlatform

//# WEAK RELATIONSHIP
1. //Shares

2. //Owns
3. //Subscribes
4. //AvailableOn
5. //PlaylistSong

#Cardinality

Cardinality of a relationship can be defined as the number of times an entity of an entity set participates in a relationship set.

1. Listener Share PlayList. (It can have more than one Listener for same Playlist)	M:N
2. Listener owns playlist. (He is only the listener and can listen more than one playlist)	1:N
3. Song available on SongOnPlatform.	M:N
4. Creator createSong CreateSong (One creator can create multiple song but same song cant be created by multiple creator)	1:N

5. Listener rates Song (one listener can rate number of songs and one song can be rated by n users)	N : M
6. Listener subscribes subscription. (subscription is unique for each user)	1:1
7. Song Playlistsong Playlist (many playlist can have common song)	M : N
8. Recommendation of song. (Only limited number of song can be determined on basis of overall popularity)	1:N
9. Song AvailabeleOn SongOnPlatform	N:M
10. Song belongsTo Movie (every movie have multiple song)	1:N
11. Song SingBY Singer (Song can be sing by multiple singers and every singer have many songs)	N:M

12. Song ComposedBY composer (every song composed by 1 composer)	N:1

→ All the relations with the schema :

1. **User (**UserID bigint, Username varchar(20), email varchar(20), Password varchar(10), Primarykey(UserID)**)**

2. **Creator(**CreatorID bigint, CreatorName varchar(20), UserID bigint,PrimaryKey(CreatorID), ForeignKey(UserID)**)**

3. **Listener(**ListnerName varchar(20), ListenerID bigint, UserID bigint, PrimaryKey(ListenerID), ForeignKey(UserID)**)**

4. **CreatorSong (** SongID bigint, creatorID bigint, Role varchar(10), PrimaryKey(SongID), ForeignKey(CreatorID)**)**

5. **Song(**SongID bigint, SongName varchar(25), ArtistName varchar(10), Lyrics varchar(3000), AlbumName varchar(10), Duration Numeric(4,2),ReleaseDate Date,Rating Numeric(3,2),countOfListener bigint,Genre varchar(10),MovieID bigint, PrimaryKey(SongID)**)**

//Subscribes (UserID char, SubscriptionID char,PrimaryKey(UserID, SubscriptionID)**)**

6. **Subscription** (PrimaryKey(User_ID char, Subscription_type varchar(20)), Price numeric)

7. **Playlist** (PlaylistID bigint, PlaylistName varchar(20), UserID bigint, Duration numeric,PrimaryKey(PlaylistID), ForeignKey(UserID))
8. **Recommendation (**RecommendationRank bigint, SongID bigint, PrimaryKey(RecommendationRank), ForeignKey(SongID)**)**

9. **Singer** (SingerID bigint, SingerName varchar(20), PrimaryKey(SingerID))

10. **Composer** (ComposerID bigint, ComposerName varchar(20), PrimaryKey(ComposerID))
11. **Song_on_Platform**(PrimaryKey(Platform_ID char,Song_ID char), Platform_link varchar(50))

12. **Movie**(Movie char, Primary Key(MovieID char, SongID char), MovieName varchar(20))

13. **Platform** (PlatformID bigint, PlatformName varchar(20), PrimaryKey(PlatformID)**)**

14. **Rates** (SongID bigint, Rating Int, ListnerID int,PrimaryKey(SongId,ListnerID))

15. **Owns** (PlaylistID bigint,UserID bigint, Ownership_Type char ,PrimaryKey(PlaylistID , UserID))
16. **PlaylistSong** (SongID bigint, PlaylistID bigint, Primary key(SongID PlaylistID))
17. **Movies** (MovieID bigint, Movie_Name varchar(20),Primary key(MovieID))
18. **Share** (UserID bigint, PlaylistID bigint, AccessType char, primarykey(UserID,PlaylistID))