Project Title: Music Streaming Service Database

System Design

1. ER Model

The **Entity-Relationship (ER) Model** will define key entities and their relationships:

Key Entities & Attributes

- **User** (*UserID, Name, Email, SubscriptionType, JoinDate*)
- **Artist** (*ArtistID, Name, Genre, Label*)
- **Song** (*SongID, Title, Duration, ReleaseDate, ArtistID, AlbumID*)
- **Album** (*AlbumID, Title, ReleaseYear, ArtistID*)
- **Playlist** (*PlaylistID, Name, UserID, CreatedDate*)
- **Subscription** (*SubscriptionID, UserID, Type, StartDate, EndDate, PaymentID*)
- **Payment** (*PaymentID, OrderID, Amount, PaymentMethod, PaymentStatus*)

2. Relationships:

- A **User** can create multiple **Playlists**
- A **Playlist** contains multiple **Songs**
- An **Artist** can release multiple **Albums**, and each **Album** contains multiple **Songs**
- A **User** can have **one active Subscription**

3. Foreign Key Relationships:

- Song(ArtistID) → Artist(ArtistID)
- Song(AlbumID) → Album(AlbumID)
- Album(ArtistID) → Artist(ArtistID)
- Playlist(UserID) → User(UserID)
- Subscription(UserID) → User(UserID)
- Subscription(PaymentID) → Payment(PaymentID), NULL if Free
- Payment(PaymentID) → Subscription(PaymentID), one-to-one

Relationship Table

Foreign Key Relationship	**Type**	**Explanation**
Song(ArtistID) → Artist(ArtistID)	One-to-Many (1:M)	One artist can have multiple songs, but a song belongs to only one artist.
Song(AlbumID) → Album(AlbumID)	One-to-Many (1:M)	One album can have multiple songs, but a song belongs to only one album (or NULL if it's a single).
Album(ArtistID) → Artist(ArtistID)	One-to-Many (1:M)	One artist can release multiple albums, but each album belongs to one artist.
Playlist(UserID) → User(UserID)	One-to-Many (1:M)	A user can create multiple playlists, but a playlist belongs to only one user.
Subscription(UserID) → User(UserID)	One-to-One (1:1)	A user can have only one active subscription at a time.
Subscription(PaymentID) → Payment(PaymentID)	One-to-One (1:1), NULLABLE	A subscription is linked to one payment (if paid), but free users won't have a payment record.
Payment(PaymentID) → Subscription(PaymentID)	One-to-One (1:1)	Each payment is tied to exactly one subscription.
Playlist-Song (Separate Join Table Needed)	Many-to-Many (M:N)	A song can be in multiple playlists, and a playlist can have multiple songs.

EntityList Table

Entity	Attribute	Data Type	Description
User	UserID	INT (Primary Key, Auto-Increment)	Unique identifier for users
	Name	VARCHAR(100)	Full name of the user
	Email	VARCHAR(255), UNIQUE	User email (used for login)
	SubscriptionType	ENUM(‘Free’, ‘Premium’)	Subscription category
	JoinDate	DATE	Date of account creation
Artist	ArtistID	INT (Primary Key, Auto-Increment)	Unique identifier for artists
	Name	VARCHAR(100)	Name of the artist
	Genre	VARCHAR(50)	Genre of music
	Label	VARCHAR(100) NULLABLE	Record label name (NULL if independent)
Song	SongID	INT (Primary Key, Auto-Increment)	Unique identifier for songs
	Title	VARCHAR(255)	Name of the song
	Duration	TIME	Duration of the song (HH:MM:SS)
	ReleaseDate	DATE	Date when the song was released
	ArtistID	INT (Foreign Key)	References Artist(ArtistID)
	AlbumID	INT (Foreign Key, NULLABLE)	References Album(AlbumID), NULL if single
Album	AlbumID	INT (Primary Key, Auto-Increment)	Unique identifier for albums
	Title	VARCHAR(255)	Name of the album
	ReleaseYear	YEAR	Year of release
	ArtistID	INT (Foreign Key)	References Artist(ArtistID)
Playlist	PlaylistID	INT (Primary Key, Auto-Increment)	Unique identifier for playlists
	Name	VARCHAR(255)	Name of the playlist
	UserID	INT (Foreign Key)	References User(UserID), playlist owner
	CreatedDate	DATE	Date of playlist creation
Subscription	SubscriptionID	INT (Primary Key, Auto-Increment)	Unique identifier for subscriptions
	UserID	INT (Foreign Key)	References User(UserID)
	Type	ENUM(‘Free’, ‘Premium’)	Subscription type
	StartDate	DATE	Subscription start date
	EndDate	DATE	Subscription expiration date
	PaymentID	INT (Foreign Key, NULLABLE)	References Payment(PaymentID), NULL for free users
Payment	PaymentID	INT (Primary Key, Auto-Increment)	Unique identifier for payments
	OrderID	VARCHAR(50), UNIQUE	Unique transaction ID
	Amount	DECIMAL(10,2)	Amount paid
	PaymentMethod	ENUM(‘Credit Card’, ‘UPI’, ‘PayPal’)	Payment type
	PaymentStatus	ENUM(‘Pending’, ‘Completed’, ‘Failed’)	Status of the payment