WolfMedia

A Media Streaming Service

CSC 540 Database Management Systems

Project Report 1

Team W

Naveen Jayanna - njayann,
Samarth Purushothaman - spurush,
Naga Srilekha Gudipati - sngudipa,
Sai Krishna Teja Varma Manthena - smanthe

February 13, 2023

1. Problem Statement

The project is to design and build a database system for WolfMedia, a media streaming service. WolfMedia is an audio and media streaming service that features songs and podcasts by various artists, and hosts. The media streaming service requires maintaining data of songs, artists, record labels, albums, podcasts, hosts and podcast episodes. The database system will be used by the administrators/management to perform various tasks such as information processing, generating royalties, payment processing and generating annual or monthly reports.

A database is a good idea for this task because it provides an organized and centralized method to store, manage, and retrieve data. A database allows for efficient and reliable storage of large amounts of data. In addition, it provides a set of powerful tools for data retrieval and analysis, which is crucial for generating reports on songs, podcasts, subscribers, listeners, and payments. A database also ensures data consistency which is essential for maintaining accuracy in sensitive operations like generating royalties and payments. Therefore, it is the ideal solution over a file system for this task for managing the data of a complex service like WolfMedia.

2. Classes of Users

a. **Payment Staff:** Payment Staff are the users that control the payments system in the application, i.e. the status of subscriptions of the users, manage the monthly payments to the record labels and podcast hosts from the royalties collected, generate the monthly payment reports, and maintain payments consistency across the entire system.

b. **Record label:** Record Labels contract multiple artists to produce albums for the application. Record Labels are responsible for paying the artists from the royalties collected from the song, where they keep 30% for themselves and distribute the rest equally to the artists of the song.

c. **Artists:** Artists hold a contract with a record label, release songs and the albums for the application. Artists can collaborate with other artists in songs. Artists get paid by the record label for the royalties collected from the song. They keep track of their monthly listeners and subscribers, generate reports on their songs and the payments they receive from the record label.

d. **Podcast Hosts:** Podcast Hosts host podcasts on the application where they are affiliated to sponsors and invite guests for their podcast episodes. They keep track of their subscribers, episode play count and ratings for the episodes, generate reports on their earnings from the episodes and advertisements.

e. **Application Users:** Application users interact with the application to listen to songs and podcasts. They pay a monthly subscription fee to the streaming service for subscribing to the artist and the podcasts. The subscribed application users can rate the podcasts.

3. Five Main Things

a. **Account Information:** Account information contains the accountID, password, the type of account, and the date of registration.

b. **Payments Information:** Payments Information comprises the various payments that happen in the system. We store the Amount paid, and the Date of Payment in each scenario of payment to record label, artist and podcast host, and the monthly subscription fee and the Start Date of Subscription.

c. **Podcast Information:** Podcast Information contains information on the host, the episodes, subscribers, guests, sponsors, and advertisements with attributes like podcast name, episode title, rating, host name, advertisement count, listening count, episode release date, duration of each episode, sponsor name and the guest name.

d. **Song Information:** Song Information includes song title, artist information, album information, the collaborators with attributes like monthly listeners for each artist, royalty rate for song, song name, play count, genre, the track number of the song, release date of the album, the main artist of the song, and collaborator name.

e. **Subscription Information:** This Information includes the user subscription to podcasts and the artists with attributes like status of Subscription, artist account ID, user account ID, podcast ID, and rating given by the subscriber to the podcast.

4. Realistic Situations:

a. A user decides to subscribe to an artist and a podcast. Monthly subscription fee is received from the subscriber to the application, and the status of the subscription is updated. He/she subscribes to the artist and the podcast, is added to the subscribers list, and can rate the podcast.

b. The record label decides to pay the artist from the royalties for the song for the month. It generates a report on earnings from the song , makes a payment to the artist from the royalties collected, and the payment is recorded in the system.

5. Assumptions

a. Every Artist/ Podcast Host/ Record Label will have accounts.
b. AccountID is the username of the account, it has to be unique.
c. Each artist is associated with exactly one record label.
d. Each song has a unique track ID within its album.
e. Each song is assigned to exactly one album (song cannot exist without an album)
f. Each song has a unique title within the album it resides.
g. Every Album belongs to a single main artist, other artists will be collaborators on the songs
h. Every Podcast will have a Podcast Host
i. Podcast episode titles are unique within a podcast.
j. Each podcast episode is assigned to exactly one podcast (podcast episodes cannot exist without a podcast).
k. Subscribers pay a flat recurring fee every month (similar to Spotify) and can subscribe to any number of artists and/or podcasts (similar to YouTube).
l. Only those who have subscribed to the WolfMedia application can access the content within it.
m. Play count for each song and episode is incremented when a WolfMedia subscriber plays it.
n. Each podcast episode is paid out to the host(s) once by WolfMedia through a one-time payment.
o. The revenue generated by a podcast episode is evenly split amongst its hosts.

6. APIs

Information Processing

- newAccount(accountID, password, typeOfAcct)
  Returns Confirmation

- updateAccount(accountID, password, typeOfAcct)
  Returns Confirmation

- deleteAccount(accountID)
  Returns Confirmation

- newUser(accountID, firstName, lastName, phoneNum, email, registrationDate, monthly PaymentAmt, statusOfSubscription)
  Returns Confirmation

- updateUser(accountID, firstName, lastName, phoneNum, email, registrationDate, monthly PaymentAmt, statusOfSubscription)

Returns Confirmation

- deleteUser(accountID)
  Returns Confirmation

- newArtist(accountID, artistName, status, type, country, primaryGenre, monthlyListeners, albums, recordLabel)
  Returns Confirmation

- updateArtist(accountID, artistName, status, type, country, primaryGenre, monthlyListeners, albums, recordLabel)
  Returns Confirmation

- deleteArtist(accountID)
  Returns Confirmation

- subscribeUserToArtist(userAccountID, artistAccountID)
  Returns Confirmation

- unsubscribeUserFromArtist(userAccountID, artistAccountID)
  Returns Confirmation

- newAlbum(albumName, artists, songs, releaseYear, edition)
  Returns albumID

- updateAlbum(albumID, albumName, artists, songs, releaseYear, edition)
  Returns Confirmation

- deleteAlbum(albumID)
  Returns Confirmation

- newSong(albumID, songTitle, mainArtist, duration, genres, playCount, releaseDate, releaseCountry, language, royaltyRate, collaborators, royaltyStatus, trackNumber)
  Returns songID

- updateSong(songID, albumID, songTitle, mainArtist, duration, genres, playCount, releaseDate, releaseCountry, language, royaltyRate, collaborators, royaltyStatus, trackNumber)
  Returns Confirmation

- deleteSong(songID, albumID)
  Returns Confirmation

- newPodcastHost(podcastHostAccountID, firstName, lastName, phoneNum, email, city)
  Returns Confirmation

- updatePodcastHost(podcastHostAccountID, firstName, lastName, phoneNum, email, city)
  Returns Confirmation

- deletePodcastHost(podcastHostAccountID)
  Returns Confirmation

- newPodcast(podcastHost, podcastName, language, country, episodeCount, genres, rating, sponsors, total subscribers)
  Returns podcastID

- updatePodcast(podcastID, podcastHost, podcastName, language, country, episodeCount, genres, rating, sponsors, total subscribers)
  Returns Confirmation

- deletePodcast(podcastID)
  Returns Confirmation

- subscribeUserToPodcast(userAccountID, podcastID)
  Returns Confirmation

- unsubscribeUserFromPodcast(userAccountID, podcastID)
  Returns Confirmation

- newPodcastEpisode(podcastID, episodeTitle, duration, releaseDate, listeningCount, specialGuests, advertisementCount)
  Returns Confirmation

- updatePodcastEpisode(podcastID, episodeTitle, duration, releaseDate, listeningCount, specialGuests, advertisementCount)
  Returns Confirmation

- deletePodcastEpisode(podcastID, episodeTitle)
  Returns Confirmation

- addSongToAlbum(albumID, songTitle, mainArtist, duration, genres, albumID, playCount, releaseDate, releaseCountry, language, royaltyRate, collaborators, royaltyStatus, trackNumber))
  Returns Confirmation

- addArtistToAlbum(artistAccountID, albumID)
  Returns Confirmation

- addArtistToRecordLabel(artistAcocuntID, recordLabelAccountID)
  Returns Confirmation

- addHostToPodcast(podcastHostAccountID, podcastID)
  Returns Confirmation

- addEpisodeToPodcast(podcastID, episodeTitle, duration, releaseDate, listeningCount, specialGuests, advertisementCount)
  Returns Confirmation

- addGenresToSong(songID, AlbumID, genres)
  Returns Confirmation

- addGenresToPodcast(podcastID, genres)
  Returns Confirmation

- addCollaboratorsToSong(songID, AlbumID, collaborators)
  Returns Confirmation

- ratePodcast(userAccountID, podcastID, rating)
  Returns Confirmation

- addSpecialGuestsToPodcastEpisode(podcastID, episodeTitle, guestID)
  Returns Confirmation

Maintaining metadata and records

- enterSongPlayCount (songTitle, albumID, playCount)
  Return confirmation

- updateSongPlayCount (songTitle, albumID, newPlays)
  Return confirmation

- enterMontlyListeners (accountID, monthlyListeners)
  Return confirmation

- updateMontlyListeners (accountID, newListeners)
  Return confirmation

- enterTotalSubscribers (podcastID, totalSubscribers)
  Return confirmation

- updateTotalSubscribers (podcastID, newSubscribers)
  Return confirmation

- enterPodcastRating (podcastID, rating)
  Return confirmation

- updatePodcastRating (podcastID, newRating)

Return confirmation

- enterListeningCount (podcastID, podcastTitle, listeners)
  Return confirmation

- updateListeningCount (podcastID, podcastTitle, newListeners)
  Return confirmation

- getSongsFromArtist (accountID)
  Returns all the songs part of the specified artist.

- getSongsFromAlbum (albumID)
  Returns all the songs part of the specified album.

- getEpisodesFromPodcast (podcastID)
  Returns all the episodes part of the specified podcast.

Maintaining payments

- makeRoyaltyPaymentToRecordLabelForSong(albumId, songTitle, month, year)
  Returns confirmation

- makePaymentToArtistsFromRecordLabel(recordLabelAccountID, albumId, songTitle, month, year)
  Returns confirmation

- makePaymentToPodcastHost(podcastHostAccountID, month, year)
  Returns confirmation

- receivePaymentFromSubscriber(userAccountID, month, year, amt)
  Returns confirmation

- getEarningsOfArtist(artistAccountID, month, year)
  Returns earnings of the artist for the given month and year

- getEarningsOfPodcastHost(podcastHostAccountID, month, year)
  Returns earnings of the podcast host for the given month and year

- getEarningsOfRecordLabel(recordLabelAccountID, month, year)
  Returns earnings of the record label for the given month and year

- getAnnualSubcriptionFeesOfUser(userAccountID, year))
  Returns total fees paid by the user in the given year

- UpdatePaymentRecordOfUser(userAccountID, month, year, amount)
  Returns confirmation

- UpdatePodcastHostPaymentRecord(podcastHostAccountID, month, year, amount)
  Returns confirmation

- UpdateRecordLabelPaymentRecord(recordLabelAccountID, month, year, playCount, amount)
  Returns confirmation

- UpdateArtistPaymentRecord(artistAccountID, month, year, amount)
  Returns confirmation

Reports

- getMonthySongPlayCount (songTitle, albumID)
  Returns the monthly play for the specified Song.

- getMonthyAlbumPlayCount (albumID)
  Returns the monthly play for the specified Album.

- getMonthyArtistPlayCount (artistID)
  Returns the monthly play for the specified Artist

- getTotalArtistPayment (artistID, startMonth, startYear, endMonth, endYear)
  Returns the total payment made out to the specified artist in the given time period.

- getTotalHostPayment (hostID, startMonth, startYear, endMonth, endYear)
  Returns the total payment made out to the specified podcast host in the given time period.

- getTotalLabelPayment (labelID, startMonth, startYear, endMonth, endYear)
  Returns the total payment made out to the specified record label in the given time period.

- getTotalRevenuePerMonth (year, month)
  Returns the total revenue of the streaming service for the given month.

- getTotalRevenuePerYear (year)
  Returns the total revenue of the streaming service for the given year

- getSongsFromArtist (artistID)
  Returns all the songs part of the specified artist.

- getSongsFromAlbum (albumID)
  Returns all the songs part of the specified album.

- getEpisodesFromPodcast (podcastID)

  Returns all the episodes part of the specified podcast.

7. Short description of views

a. **Payment Staff:** The payment staff can access all payment tables and subscriptions in the system. They oversee payments and generate reports to obtain statistics about revenues or marketing design. They can access all payment details and update anything to make corrections.

b. **Artists:** Artists can view their albums, songs, the record label they are associated with and track the money they've earned from his/her music. They can release new albums.

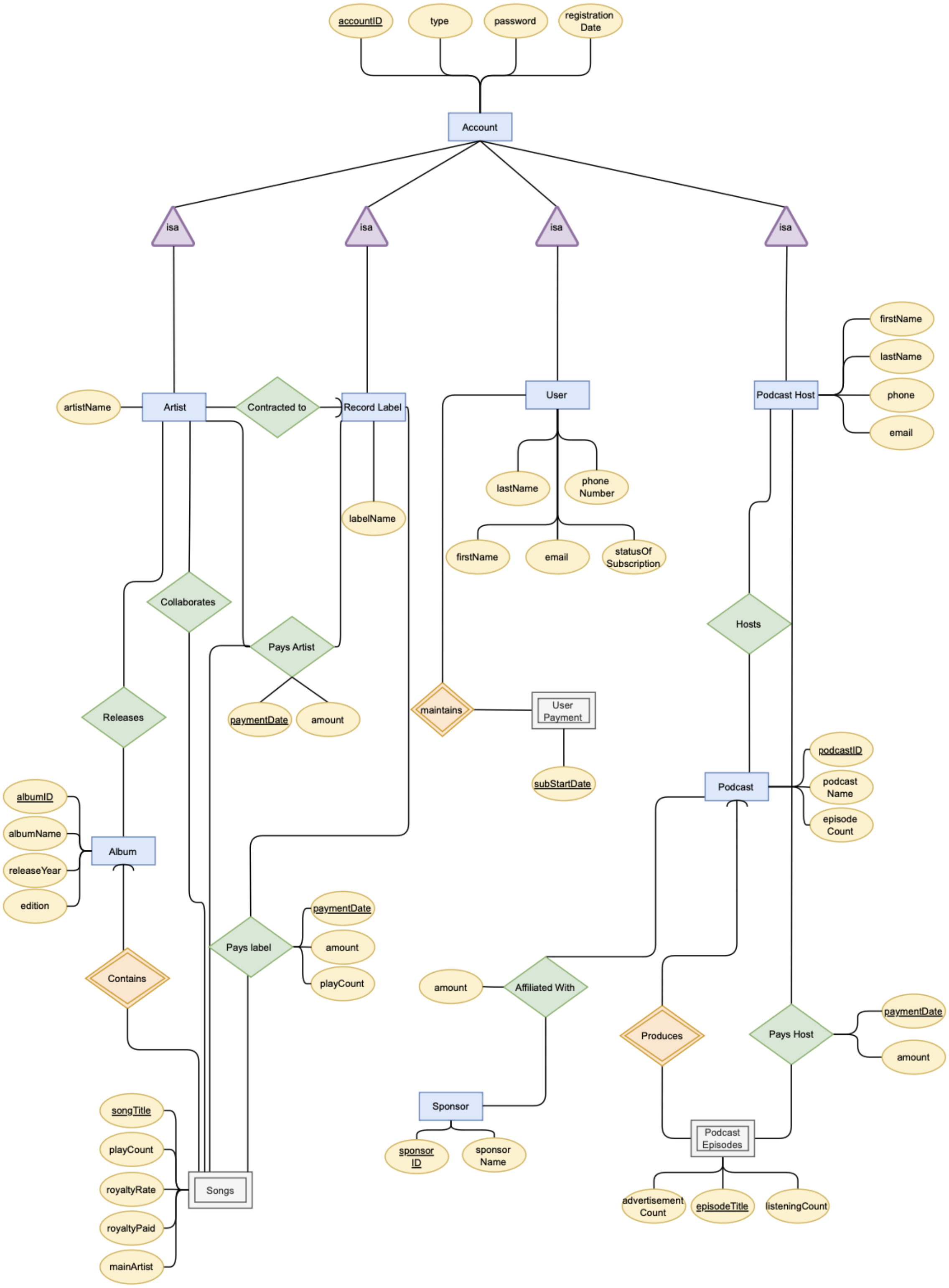
c. **Podcast Hosts:** Podcast hosts can view their podcasts and the corresponding episodes. They can see the podcast's ratings and the money every episode makes.

d. **Users:** Users can give ratings to podcasts, subscribe/unsubscribe from an artist or a podcast and can view their payment history. They cannot view podcast sponsor information or song royalty details.

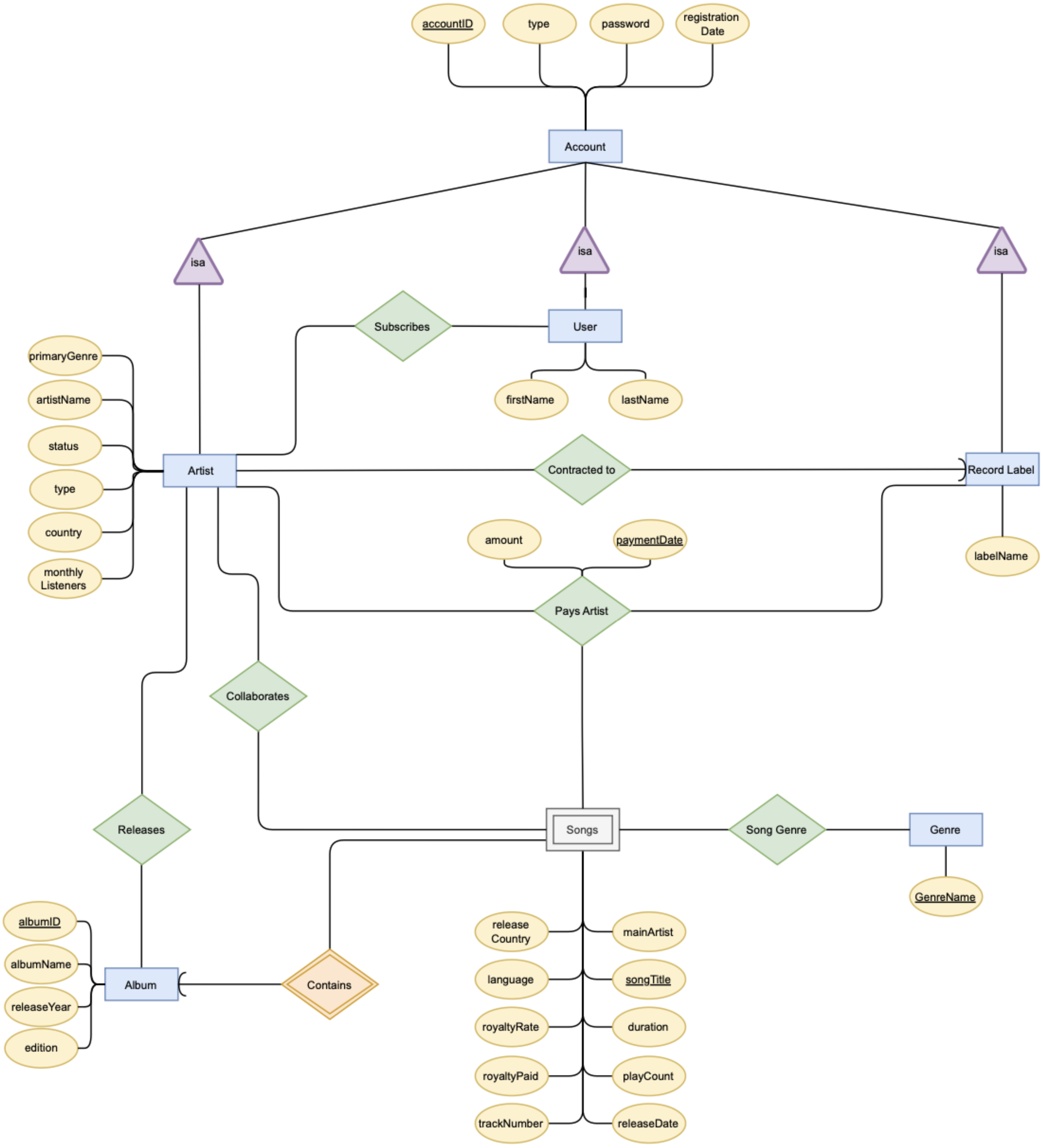
e. **Record label:** Record labels can contract new artists, view their earnings and distribute these earnings to artists. They can see statistics of their payment history.

8. Local E/R diagram

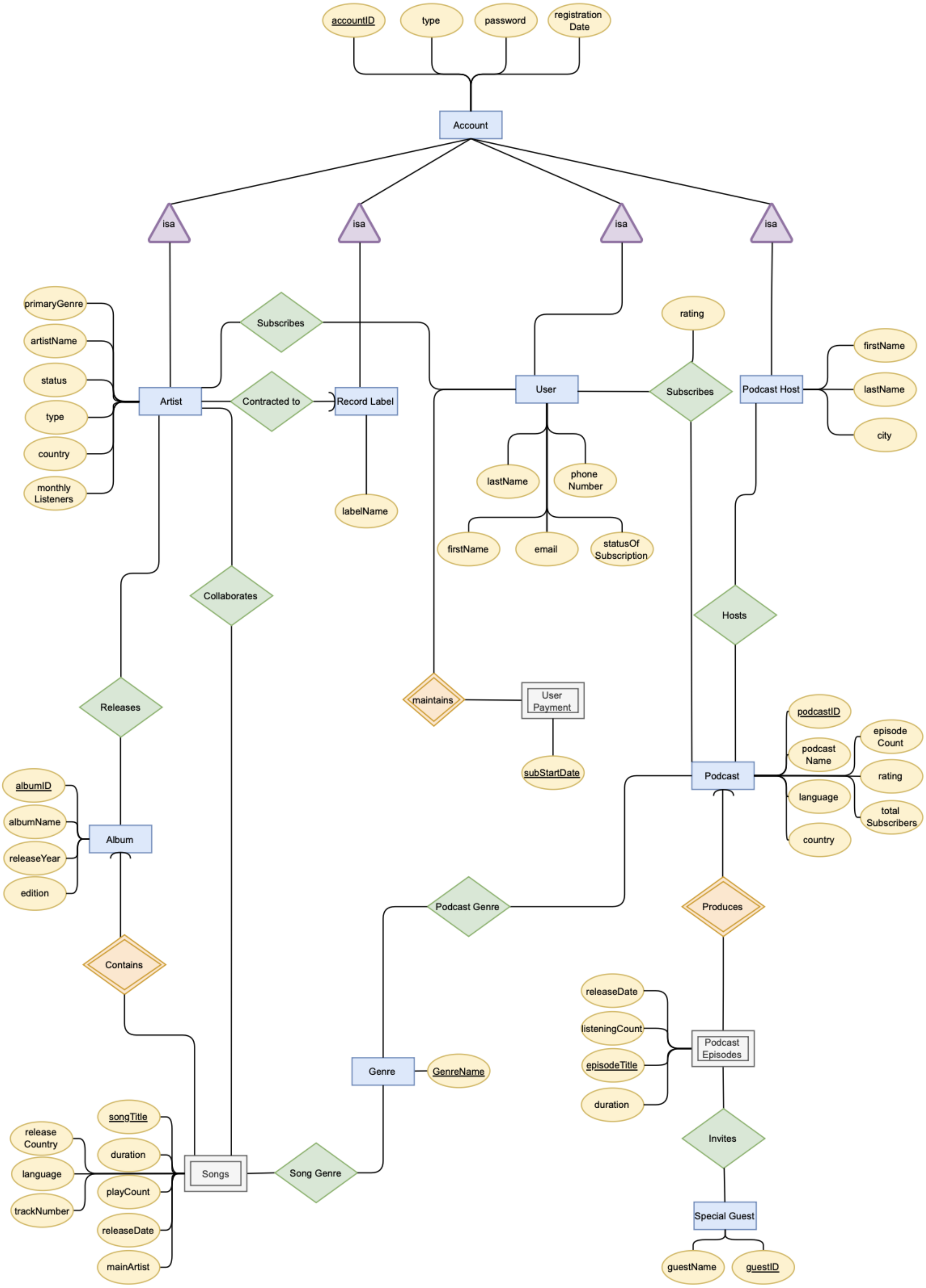
Payment Staff View



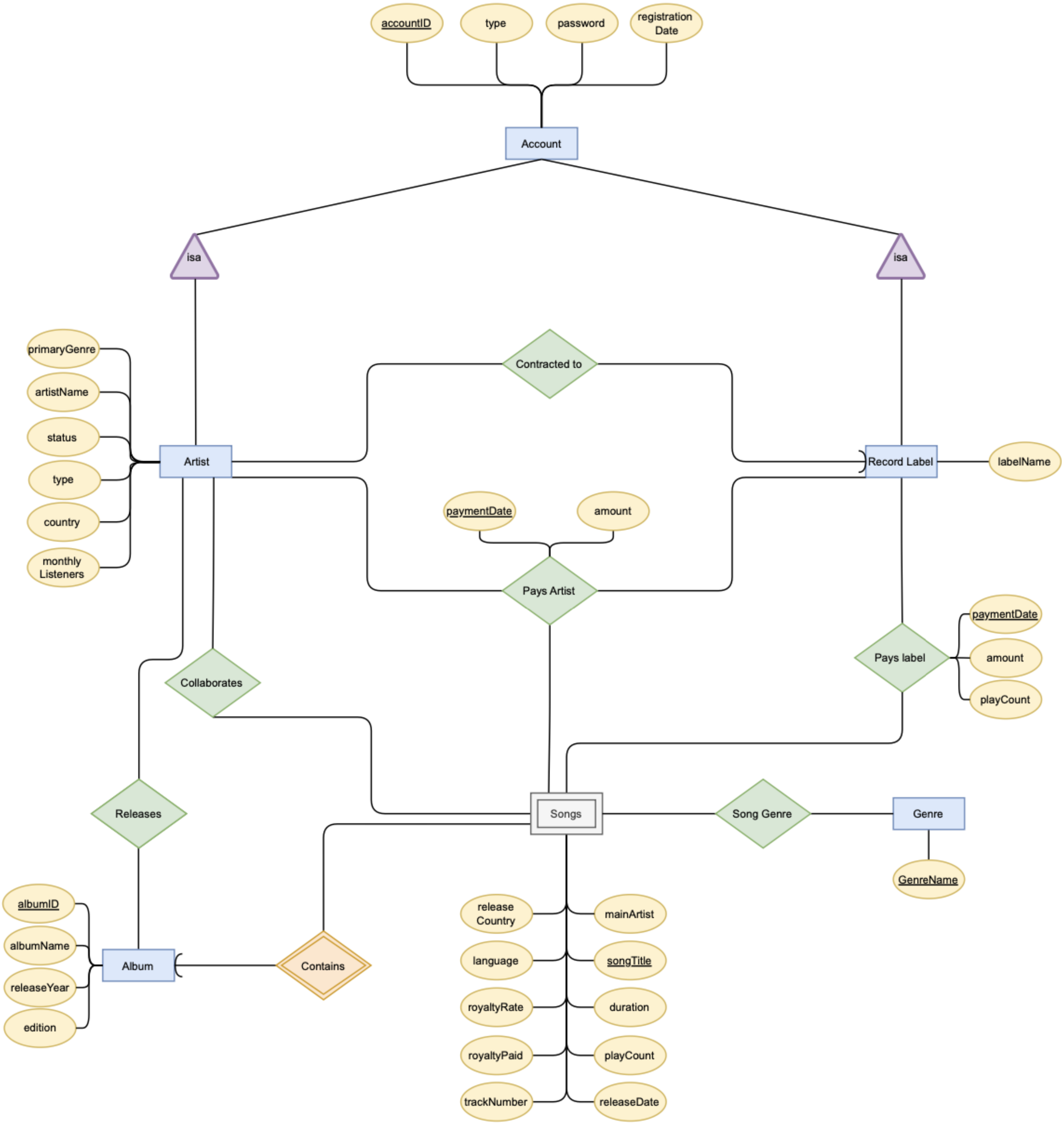
Artist View


accountID
type
password
registration Date
Account
isa
isa
isa
User
Subscribes
firstName
lastName
primaryGenre
artistName
status
type
country
monthly Listeners
Artist
Contracted to
Record Label
labelName
amount
paymentDate
Pays Artist
Collaborates
Releases
Songs
Song Genre
Genre
GenreName
albumID
albumName
releaseYear
edition
Album
Contains
release Country
language
royaltyRate
royaltyPaid
trackNumber
mainArtist
songTitle
duration
playCount
releaseDate

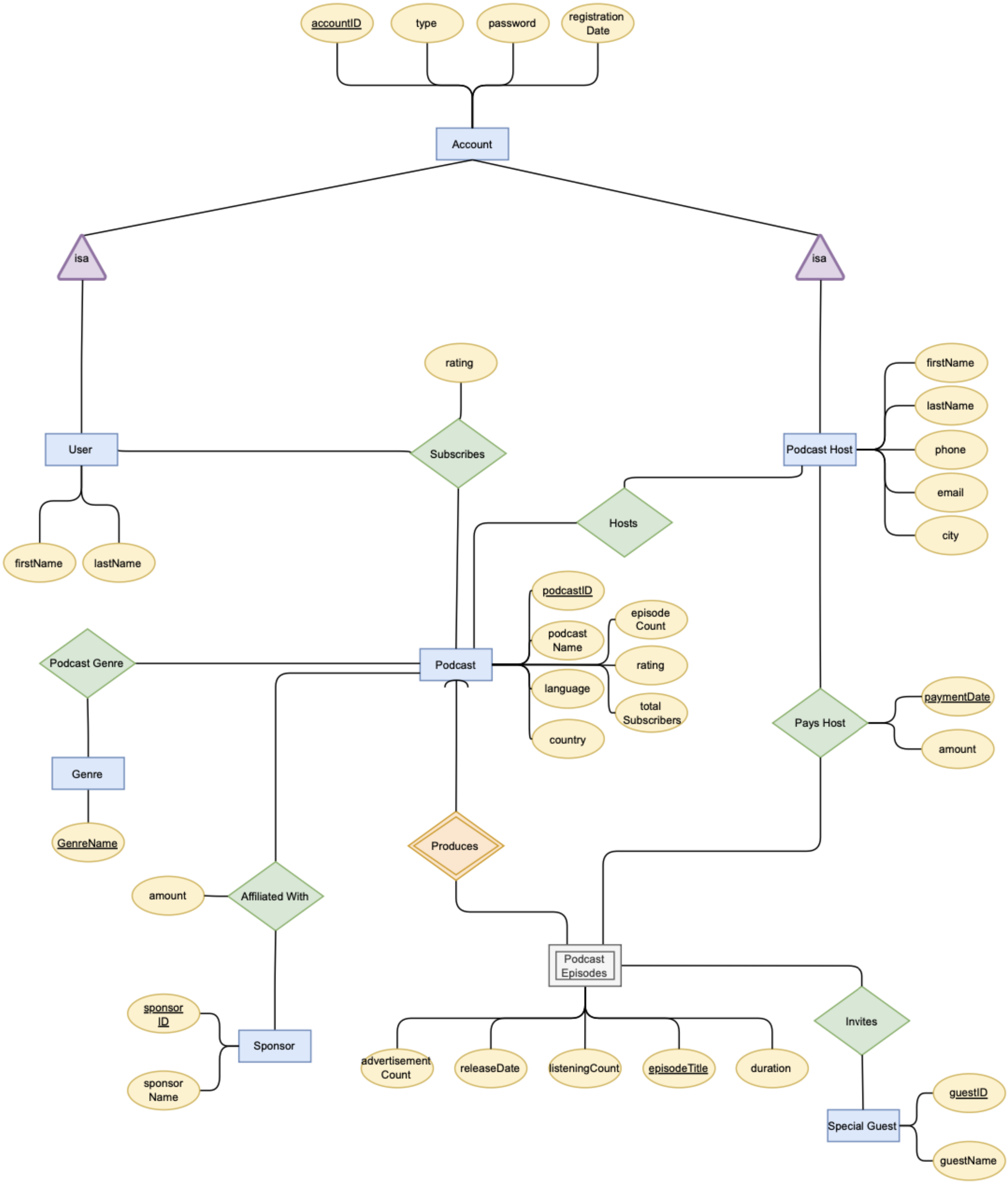
User View


accountID
type
password
registration Date
Account
isa
isa
isa
isa
primaryGenre
artistName
status
type
country
monthly Listeners
Artist
Subscribes
Contracted to
Record Label
labelName
User
rating
Subscribes
Podcast Host
firstName
lastName
city
lastName
phone Number
firstName
email
statusOf Subscription
Collaborates
Hosts
Releases
maintains
User Payment
subStartDate
podcastID
podcast Name
episode Count
rating
language
total Subscribers
country
Podcast
albumID
albumName
releaseYear
edition
Album
Podcast Genre
Produces
Contains
releaseDate
listeningCount
episodeTitle
duration
Podcast Episodes
Genre
GenreName
Invites
songTitle
release Country
duration
language
Songs
Song Genre
playCount
trackNumber
releaseDate
mainArtist
Special Guest
guestName
guestID

Record Label View



Podcast Host View



9. Local E/R diagrams Description

a. The Account is subdivided into User, Artist, Podcast Host and Record Label. This will allow a holistic representation of all the users in this problem statement and allow them to access everything they need based on the type of account they use.

b. The User, Artist, Podcast Host and Record Label will inherit the Account ID as the tables' respective primary keys.

c. The song titles can be the same in different albums, making the song a weak entity. The primary key for the Songs table will be (albumID, songTitle).

d. Similarly, the podcast episode is a weak entity as episode titles can be the same in different podcasts. The primary key for the Podcast Episodes table will be (podcastID, episodeTitle)

e. Each Artist will be contracted by exactly one Record Label. Hence the rounded arrow from Artist to Record Label.

f. Every Song should belong to exactly one Album and every Podcast Episode should belong to exactly one Podcast. Therefore, both the relations also have a rounded arrow to capture this characteristic.

g. Genres and Special Guests are separate entities because they are a multi-valued attribute on the Song Entity and the Podcast Episode entity respectively.

h. Podcasts also have a rating field, which can be given by users who subscribed to the podcast, hence it can be captured by the Subscribed Podcast Relationship.

i. Pays Artist is a triple entity connecting relation as the Record Label pays the Artists per song basis.

j. User Payment is a weak entity by itself. As the monthly subscription fee is a constant, we make entries in this table for months for which the user pays subscription fee to WolfMedia. This is why date is the only attribute on this table, its primary key will have the user account ID.

10. Local Relational Schemas

Payment Staff view

a. Account (accountID, type, password, registrationDate)

b. Artist (artistAccountID, artistName, recordLabelAccountID)

c. Releases (artistAccountID, albumID)

d. Record label (recordLabelAccountID, labelName)

e. Album (albumID, albumName, releaseYear, edition)

f. Song (albumID, songTitle, playCount, royaltyRate, royaltyPaid)

g. Collaborators (artistAccountID, songTitle, albumID)

h. Podcast (podcastID, podcastName, episodeCount)

i. Podcast host (podcastHostAccountID, firstName, lastName, phone, email)

j. Hosts (podcastHostAccountID, podcastID)
k. Podcast episode (podcastID, episodeTitle, listeningCount, advertisementCount)
l. User (userAccountID, firstName, lastName, phoneNumber, email, statusOfSubscription)
m. Sponsor (sponsorID, sponsorName)
n. Affiliated with (sponsorName, podcastID, amount)
o. User payment (userAccountID, subStartDate)
p. Pays label (songTitle, albumID, recordLabelAccountID, paymentDate, playCount, amount)
q. Pays artist (artistAccountID, recordLabelAccountID, songTitle, albumID, paymentDate, amount)
r. Pays host (podcastHostAccountID, podcastID, episodeTitle, paymentDate, amount)

Artist's view

a. Account (accountID, type, password, registrationDate)
b. Artist (artistAccountID, artistName, status, type, country, primaryGenre, monthlyListeners, recordLabelAccountID)
c. Releases (artistAccountID, albumID)
d. Record label (recordLabelAccountID, labelName)
e. Album (albumID, albumName, releaseYear, edition)
f. Genre (genreName)
g. Song genre (genreName, songTitle, albumID)
h. Song (albumID, songTitle, duration, playCount, releaseDate, releaseCountry, language, royaltyRate, royaltyPaid, trackNumber)
i. Collaborates (artistAccountID, songTitle, albumID)
j. Subscribed artist (artistAccountID, userAccountID)
k. User (userAccountID, firstName, lastName)
l. Pays artist (artistAccountID, recordLabelAccountID, songTitle, albumID, paymentDate, amount)

Podcast Host's view

a. Account (accountID, type, password, registrationDate)
b. Genre (genreName)
c. Subscribed podcast (podcastID, userAccountID, rating)
d. Podcast (podcastID, podcastName, language, country, episodeCount, rating, totalSubscribers)
e. Podcast host (podcastHostAccountID, firstName, lastName, phone, email, city)
f. Hosts (podcastHostAccountID, podcastID)
g. Podcast episode (podcastID, episodeTitle, duration, releaseDate, listeningCount, advertisementCount)
h. Podcast genre (podcastID, genreName)
i. User (userAccountID, firstName, lastName)
j. Sponsor (sponsorID, sponsorName)

k. Affiliated with (sponsorName, podcastID, amount)
l. Pays host (podcastHostAccountID, podcastID, episodeTitle, paymentDate, amount)
m. Special guest (guestID, guestName)
n. Invites (guestID, podcastID, episodeTitle)

Record Label's view

a. Account (accountID, type, password, registrationDate)
b. Artist (artistAccountID, artistName, status, type, country, primaryGenre, monthlyListeners, recordLabelAccountID)
c. Releases (artistAccountID, albumID)
d. Record label (recordLabelAccountID, labelName)
e. Album (albumID, albumName, releaseYear, edition)
f. Genre (genreName)
g. Song genre (genreName, songTitle, albumID)
h. Song (albumID, songTitle, duration, playCount, releaseDate, releaseCountry, language, royaltyRate, royaltyPaid, trackNumber)
i. Collaborators (artistAccountID, songTitle, albumID)
j. Subscribed artist (artistAccountID, userAccountID)
k. Pays label (songTitle, albumID, recordLabelAccountID, paymentDate, playCount, amount)
l. Pays artist (artistAccountID, recordLabelAccountID, songTitle, albumID, paymentDate, amount)

User view

a. Account (accountID, type, password, registrationDate)
b. Artist (artistAccountID, artistName, status, type, country, primaryGenre, monthlyListeners, recordLabelAccountID)
c. Releases (artistAccountID, albumID)
d. Record label (recordLabelAccountID, labelName)
e. Album (albumID, albumName, releaseYear, edition)
f. Genre (genreName)
g. Song genre (genreName, songTitle, albumID)
h. Song (albumID, songTitle, duration, playCount, releaseDate, releaseCountry, language, trackNumber)
i. Collaborators (artistAccountID, songTitle, albumID)
j. Subscribed artist (artistAccountID, userAccountID)
k. Subscribed podcast (podcastID, userAccountID, rating)
l. Podcast (podcastID, podcastName, language, country, episodeCount, rating, totalSubscribers)
m. Podcast host (podcastHostAccountID, firstName, lastName, city)
n. Hosts (podcastHostAccountID, podcastID)
o. Podcast episode (podcastID, episodeTitle, duration, releaseDate, listeningCount)
p. Podcast genre (podcastID, genreName)

q. User (userAccountID, firstName, lastName, phoneNumber, email, statusOfSubscription)

r. User payment (userAccountID, subStartDate)

s. Special guest (guestID, guestName)

t. Invites (guestID, podcastID, episodeTitle)

11. Documentation of Local Relational Schemas

a. Entity Sets to Relations:

i. The entity sets in our diagram were made into relations, with all their attributes the same for Account, Album, Songs, Podcast, Podcast Episodes, Genre, User Payment, Special Guest and Sponsor.

ii. The Entity sets that are subsets of Account were made into relations based on the E/R translation to avoid redundancy and save storage space.

b. Combining Many-One Relationships:

i. Every Artist will be contracted to exactly one Record Label, hence the “contracted to” table will be redundant and the Record Label is added as an attribute in the Artist table.

ii. ‘Songs’ is a weak entity, therefore, our schema will not have translation for the ‘Contains’ relationship as it is redundant.

iii. ‘Podcast Episodes’ is a weak entity, therefore, our schema will not have translation for the ‘Produces’ relationship as it is redundant.

iv. ‘User Payment’ is also a weak entity because it cannot be identified just by the payment date. It’s key will have the user_account_id from the Account table. Hence translation of ‘maintains’ relationship is redundant.

c. Relationships to Relations:

i. Relationships Subscribed Artist, Subscribed Podcast, Collaborates, Releases, Hosts, Song Genre, Podcast Genre, Invites, Affiliated with, Pays Artist, Pays Label and Pays Host from the E/R diagrams have each been turned into relations in our schema. Their keys in the schema are the combined keys of the entities they represent.

ii. Pays Artist’s key will be from three tables i.e, Songs, Artist and Record Label as the record label pays artists per song.

iii. All the relationships from weak entities to normal entities like Collaborates, Pays Artist, Pays label, Song Genre, Pays Host, Invites will have 3 keys that make up the primary key (2 attributes are the weak entity identifiers and 1 attribute is the normal entity identifier). They also have all the other attributes the relationship carries.