Valeria Cordova Mulvany & Wen Shan Jiang

Users & User Stories

1. **Artist**: approved artists that upload at least one song
2. **Listener**: listeners of all account types that have an account with Spotify. Include both paid users with all types of Spotify premium plans (Individual, Duo, Family, and Students) and unpaid users.
3. **Advertiser**: people who created at least one ad with Ad Studio, Spotify's self-serve ad platform

ID	Simple/ Complex/ Analytical	As a <role>	I want <goal>	So that <reason>
US1	Analytical	Artist	See the country that my music is streamed the most	I can better understand my fans' demographics and cater my music to those target audience
US2	Simple	Artist	See the number of followers I have	I can see how many people are constantly keeping up with my music
US3	Complex	Artist	Upload new song on Spotify	My listeners can enjoy my new music
US4	Analytical	Listener	Discover the most popular songs in a given country	I can find new music to listen to
US5	Simple	Listener	See the total number of songs in a given playlist	I have an idea of how many songs the playlist contains and how big the playlist is
US6	Simple	Listener	Download songs I like	I can listen to music without internet connection
US7	Analytical *(not yet implemented)*	Listener	Most played song of all time	I can be up to date with the trend

US8	Complex	Listener	organize my songs on a playlist by album alphabetical order	I can choose to listen to the songs in the order that I want
US9	Complex	Advertiser	See how many people have listened to my advertisement	Measure how effective my current marketing strategy is
US10	Complex	Advertiser	See the demographics of all the listeners who received our advertisements	I can see where we need to focus our efforts, see if we're reaching our target audience, and/or change marketing strategies

The Conceptual Model

Link: **https://lucid.app/invitations/accept/fe6536ee-2549-46b4-957f-4b3879984a6e**
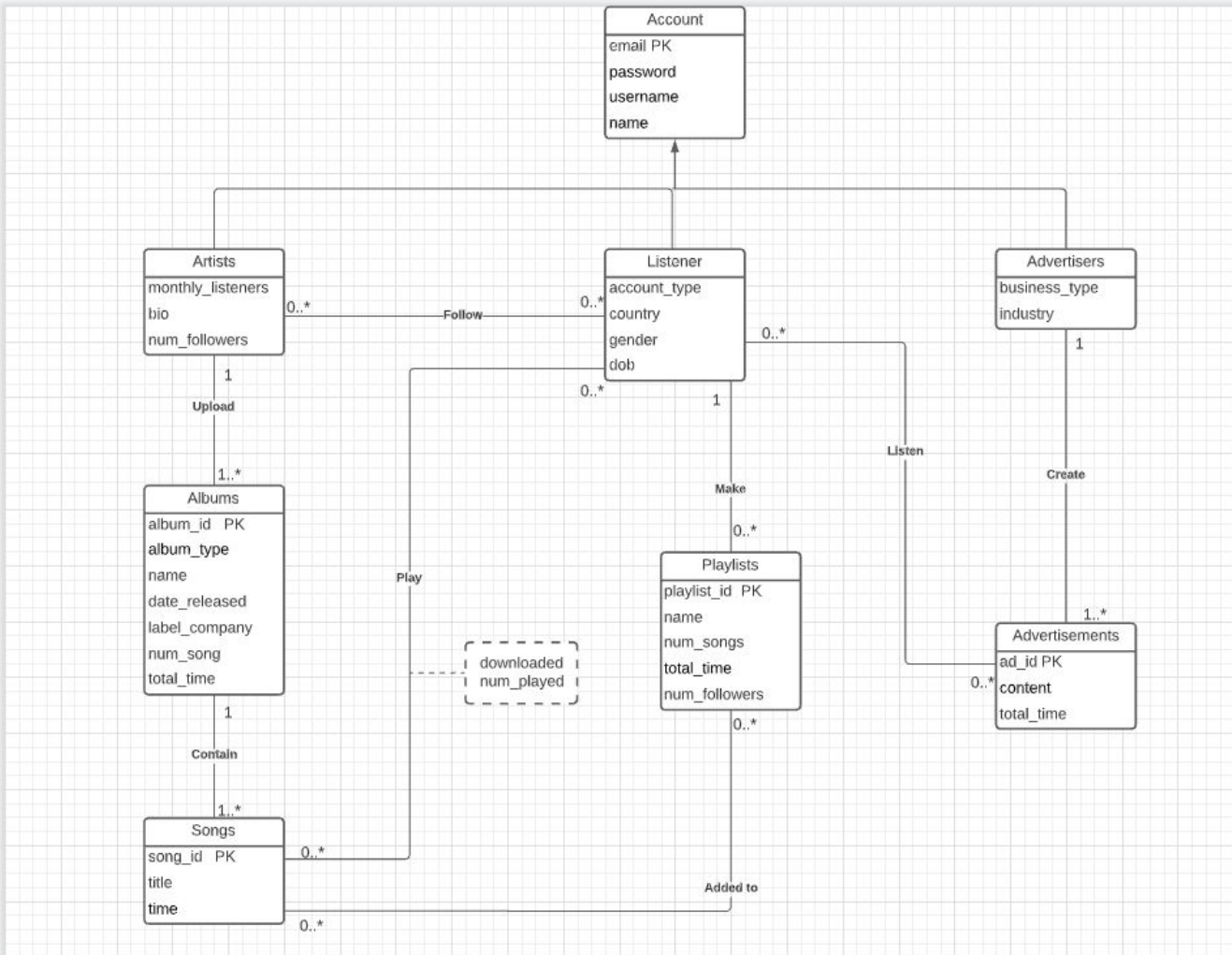
Assumptions:

1. A song can only belong to one album
2. An album can only belong to one artist
3. A playlist can only belong to one listener
4. A listener can create an empty playlist
5. An advertisement can reach zero or more people
6. An artist must upload at least one album
7. An advertiser must create at least one advertisement

Relational Model

Account (**email**, password, username, name)

Artists (**artist_email**, monthly_listeners, bio, num_followers)

Listener (**listner_email**, country, gender, dob, account_type)

Advertisers (**advertiser_email**, business_type, industry)

Albums (**album_id**, album_type, name, date_released, label_company, num_song, total_time, artist_email)

Songs (**song_id**, title, time, album_id)

Playlists (**playlist_id**, name, num_songs, total_time, num_followers, listener_email)

Advertisements (**ad_id**, content, total_time, advertiser_email)

Follow (**artist_email, listener_email**)

Play (**song_id, listener_email**, downloaded, num_played)

Added to (**song_id, playlist_id**)

Listen (**listener_email, ad_id**)

Upload: Absorbed by adding FK artist_email to entity Albums

Contain: Absorbed by adding FK album_id to entity Songs

Make: Absorbed by adding FK listener_email to entity Playlists

Create: Absorbed by adding FK advertiser_email to entity Advertisements

Functional Dependencies

Account: email → password, username, name

Artists: artist_email → monthly_listeners, bio, num_followers

Listener: listener_email →country, gender, dob, account_type

Advertisers: advertiser_email → business_type, industry

Albums: album_id → album_type, name, date_released, label_company, num_song, total_time, artist_email

Songs: song_id → title, time, album_id

Playlists: playlist_id → name, num_songs, total_time, num_followers, listener_email

Advertisements: ad_id → content, total_time, advertiser_email

Follow: artist_email, listener_email → artist_email, listener_email

Play: song_id, listener_email → downloaded, num_played

Added to: song_id, playlist_id → song_id, playlist_id

Listen: listener_email, ad_id → listener_email, ad_id

Normalization

Account: {email}+ → {email, password, username, name}

Artists: {artist_email}+ → {artist_email, monthly_listeners, bio, num_followers}

Listener: {listener_email}+ → {listener_email, account_type, country, gender, dob}

Advertisers: {advertiser_email}+ → {advertiser_email, business_type, industry}

Albums: {album_id}+ → {album_id, album_type, name, date_released, label_company, num_song, total_time, artist_email}

Songs: {song_id}+ → {song_id, title, time, album_id}

Playlists: {playlist_id}+ →{playlist_id, name, num_songs, total_time, num_followers, listener_email}

Advertisements: {ad_id}+ → {ad_id, content, total_time, advertiser_email}

Follow: {artist_email, listener_emai}+l → {artist_email, listener_email}

Play: {song_id, listener_email}+ → {song_id, listener_email, downloaded, num_played}

Added to: {song_id, playlist_id}+ → {song_id, playlist_id}

Listen: {listener_email, ad_id}+ → {listener_email, ad_id}

Each relation is already in 3NF because there are no repeating groups, partial dependencies on composite primary keys, or transitive dependencies when we designed the conceptual model. Then we also checked and there are no bad functional dependencies that needed to be resolved leading this to be in BCNF.

Vertabelo: https://my.vertabelo.com/doc/4SKCaM6Z8ThzjynXaArhpyi054L4WRz2