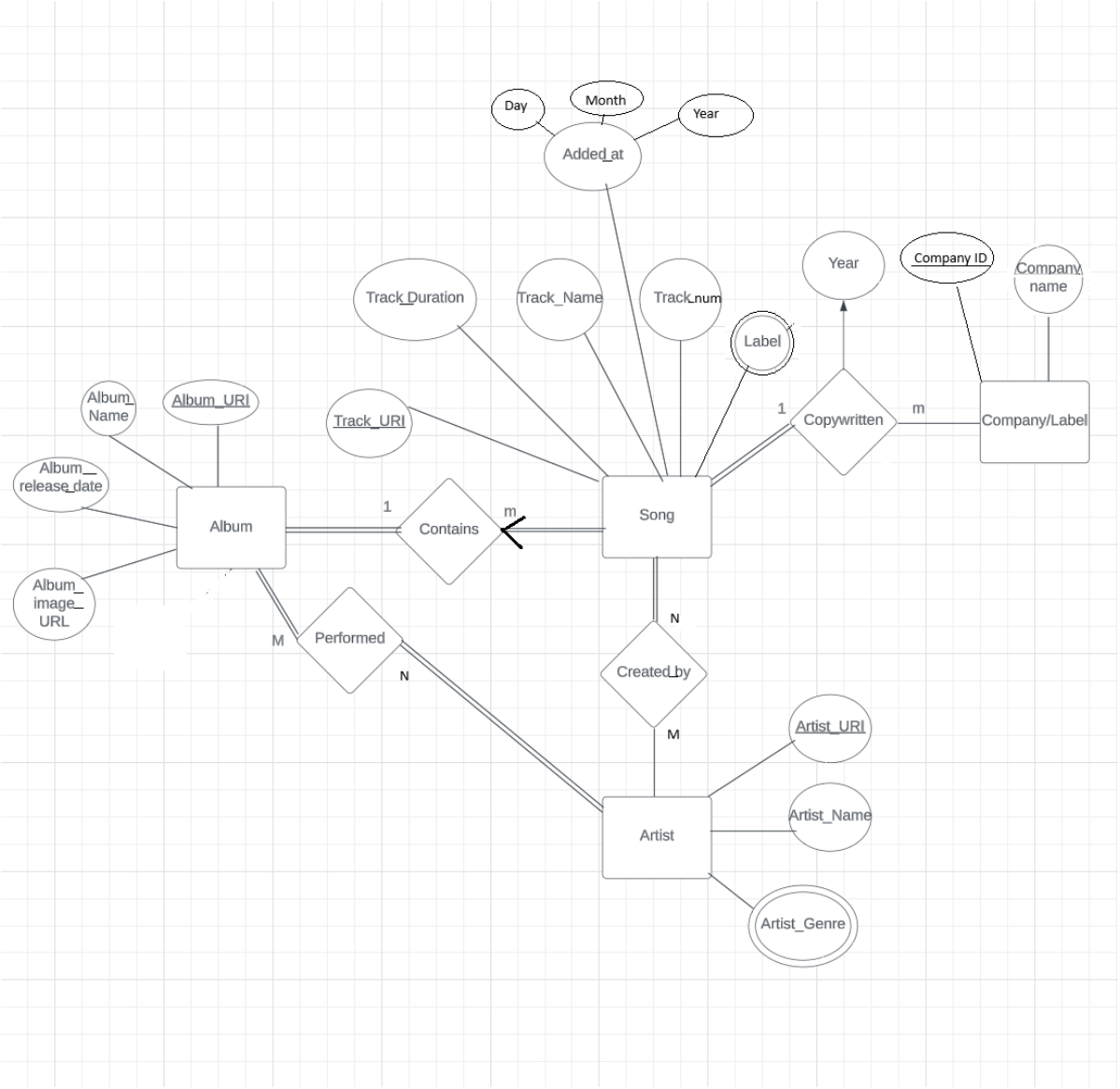

Part 1, Data Discovery:

The dataset we have chosen is a list of the top 10000 songs on Spotify obtained from Kaggle, from the 1960's till now. Each row which represents a song specifies many details such as the danceability, the artists who participated in creating said song, labels, and which album the song itself belongs to among many other details. The dataset has 10000 entries.

Normalisation:

0NF:
Song(<u>Track_URI</u>, <u>Album_URI</u>, Artist_URI, Added_at, Track_Num, Track_Duration, Labels, Album_Name, Album_release_date, Album_Artist_URI's, Album_image_URL, Copyright, Artist_Name, Artist_Genres)

1NF:

Song(Track_URI, Album_URI, Track_Num, Track_Duration, Album_Name, Album_image_URL, Artist_Name, Copyright_year)
Artists(Track_URI, Artist_URI)
ArtistGenre(Arist_URI, Genre)
Track_Copyright(Track_URI, Company_ID)
Copyright(Company_ID, Company_name)
Track_Added_On(Track_URI, year, month, day, time)
Album_Artists(Album_URI, Artist_URI)
Album_Release_Date(Album_URI, year, month, day)
Labels(Track_URI, label)

For getting from 0NF to 1NF, we broke down all compound and multi-valued attributes into their own tables, with the compound and multi-valued attributes specifically being:

- **Copyright -> Company_ID's, Company_name's, Copyright_year (Specific to song)**
- **Labels: A song can have multiple labels**
- **Artist_Genres: An artist can have multiple genres associated with them**
- **Added_at: Made up of a day, month, and a year**
- **Album_release_date: Also made up of a day, month, and a year**
- **Company_ID's: Multiple companies per song, so create a table for determining name from a Company_ID, and create support table for all Company_ID's associated with a song**
- **Album_Artist_URI's gets its own table because an album can have multiple artists associated with it**

2NF:
Song(Track_URI, Track_Num, Track_Duration, Copyright_year)
Album(Album_URI, Album_Name, Album_image_URL)
Artist(Artist_URI, Artist_Name)
Artists_Track(Track_URI, Artist_URI)
ArtistGenre(Arist_URI, Genre)
Track_Copyright(Track_URI, Company_ID)
Copyright(Company_ID, Company_name)
Album_Artists(Album_URI, Artist_URI)
Album_Release_Date(Album_URI, year, month, day)
Labels(Track_URI, label)

For getting from 1NF to 2NF, we removed partial dependency by separating Album_URI, Album_Name, and Album_image_URL into its own table, and by separating Artist_URI and Artist_Name into its own table.

3NF:

Song(Track_URI, Track_Num, Track_Duration, Copyright_year)
Album(Album_URI, Album_Name, Album_image_URL)
Artist(Artist_URI, Artist_Name)
Artists_Track(Track_URI, Artist_URI)

ArtistGenre(Arist_URI, Genre)
Track_Copyright(Track_URI, Company_name)
CopyRight(Company_ID, Company_name)
Track_Added_On(Track_URI, year, month, day, time)
Album_Artists(Album_URI, Artist_URI)
Album_Release_Date(Album_URI, year, month, day)
Labels(Track_URI, label)

BCNF: Since all of our determinants are superkeys, we are also in BCNF:

Song(Track_URI, Track_Num, Track_Duration, Copyright_year)
Album(Album_URI, Album_Name, Album_image_URL)
Artist(Artist_URI, Artist_Name)
Artists_Track(Track_URI, Artist_URI)
ArtistGenre(Arist_URI, Genre)
Track_Copyright(Track_URI, Company_name)
CopyRight(Company_ID, Company_name)
Track_Added_On(Track_URI, year, month, day, time)
Album_Artists(Album_URI, Artist_URI)
Album_Release_Date(Album_URI, year, month, day)
Labels(Track_URI, label)

Justification for participation/cardinality constraints:

- **Song -> Label: A company can have/own many songs and a song could be created by many companies that collaborate to create a song. Therefore there is a many to many relationship. A song has to be owned by a company and not every company needs to own a song, and thus, we have total participation from the song and not from the company.**
- **Songs -> Album: Every song needs to be in an album and every album needs to have at least one song in it. Thus, there is total participation on both sides between the two. Now, in an album, there could be many songs but one song can only be in one album. Thus, there is a many to one relationship between the two. And the song always determines the album it is in.**
- **Album -> Artist: In an album there could be many artists performing in them and and Artist can perform in multiple albums. Therefore there is a many to many relationship. Now for the participation, an artist must perform in an album and the album must have at least one artist performing in it. This ensures a total participation.**
- **Artist -> Song: A song has to be created by an artist and artist need to be be in every song, so there is a total participation from the song and not he artist. A song can have many artist and so can many artist create many songs, thus there is a many to many relationship between the two.**

References

Top 10000 Songs on Spotify 1960-Now. (2023, July). Kaggle. Retrieved October 15, 2023, from https://www.kaggle.com/datasets/joebeachcapital/top-10000-spotify-songs-1960-now