Data and setup

In [2]:
```
import pandas as pd
import sqlite3
from sqlalchemy import create_engine
spotify_df = pd.read_csv("F:\spotify_songs.csv")
```

Normalize data into First Normal Form

- A primary key (a unique, non-null column identifying each row)
- No repeating groups of columns
- Each cell contains a single value

In [3]:
```
spotify_df.columns
```

Out[3]:
```
Index(['track_id', 'track_name', 'track_artist', 'track_popularity',
       'track_album_id', 'track_album_name', 'track_album_release_date',
       'playlist_name', 'playlist_id', 'playlist_genre', 'playlist_subgenre',
       'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness',
       'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo',
       'duration_ms'],
      dtype='object')
```

In [4]:
```
# track_id might be a good contender for a primary key
# Check if length of unique track_id values is equal to number of rows
spotify_df.shape[0] == len(spotify_df["track_id"].unique())
```

Out[4]:
```
False
```

In [5]:
```
duplicated_track_id = (spotify_df.loc[spotify_df.duplicated(subset=['track_id'])]["track_id"].iloc[0])
# Inspect rows of duplicated track_id
spotify_df.loc[spotify_df["track_id"] == duplicated_track_id]
```

Out[5]:

	track_id	track_name	track_artist	track_popularity	track_album_id	track_album_name	track_album_release_date
739	1HfMVBKM75vxSfsQ5VefZ5	Lose You To Love Me	Selena Gomez	93	3tBkjgxDqAwss76O1YHsSY	Lose You To Love Me	2019-10-2
1299	1HfMVBKM75vxSfsQ5VefZ5	Lose You To Love Me	Selena Gomez	93	3tBkjgxDqAwss76O1YHsSY	Lose You To Love Me	2019-10-2
18320	1HfMVBKM75vxSfsQ5VefZ5	Lose You To Love Me	Selena Gomez	93	3tBkjgxDqAwss76O1YHsSY	Lose You To Love Me	2019-10-2
19730	1HfMVBKM75vxSfsQ5VefZ5	Lose You To Love Me	Selena Gomez	93	3tBkjgxDqAwss76O1YHsSY	Lose You To Love Me	2019-10-2
21555	1HfMVBKM75vxSfsQ5VefZ5	Lose You To Love Me	Selena Gomez	93	3tBkjgxDqAwss76O1YHsSY	Lose You To Love Me	2019-10-2
23641	1HfMVBKM75vxSfsQ5VefZ5	Lose You To Love Me	Selena Gomez	93	3tBkjgxDqAwss76O1YHsSY	Lose You To Love Me	2019-10-2
30388	1HfMVBKM75vxSfsQ5VefZ5	Lose You To Love Me	Selena Gomez	93	3tBkjgxDqAwss76O1YHsSY	Lose You To Love Me	2019-10-2

7 rows × 23 columns

In [6]:
```
# track_id is redundant here because it's on multiple playlists.
# We split the data frame into two tables (track and playlist)
track_columns = ['track_id',
                 'track_name',
                 'track_artist',
                 'track_popularity',
                 'track_album_id',
                 'track_album_name',
                 'track_album_release_date',
                 'danceability',
                 'energy',
                 'key',
                 'loudness',
                 'mode',
                 'speechiness',
                 'acousticness',
                 'instrumentalness',
                 'liveness',
                 'valence',
                 'tempo',
                 'duration_ms']

playlist_columns = ['playlist_id',
                    'playlist_name',
                    'playlist_genre',
                    'playlist_subgenre']

# Create a dictionary of 2 dataframes (track_df and playlist_df)
spotify_df_dict = {'track' : spotify_df.loc[:,track_columns].drop_duplicates(),
                   'playlist' : spotify_df.loc[:,playlist_columns].drop_duplicates()}
```

Now let's confirm that track_id and playlist_id are unique and non-null within each DataFrame:

In [7]:
```
# Check track_id is unique per row in track df
print("TRACK:")
print("Is there a unique id per row?")
print(spotify_df_dict["track"].shape[0] == len(spotify_df_dict["track"]["track_id"].unique()))

# Check track_id has no NA values
print("Are there NA values?")
print(spotify_df_dict["track"]["track_id"].isnull().any())

# Check playlist_id is unique per row in playlist df
print("\nPLAYLIST:")
print("Is there a unique id per row?")
print(spotify_df_dict["playlist"].shape[0] == len(spotify_df_dict["playlist"]["playlist_id"].unique()))

# Check playlist_id has no NA values
print("Are there NA values?")
print(spotify_df_dict["playlist"]["playlist_id"].isnull().any())
```

```
TRACK:
Is there a unique id per row?
True
Are there NA values?
False

PLAYLIST:
Is there a unique id per row?
False
Are there NA values?
False
```

Seems like playlist_id still isn't unique. Let's look more closely at the rows where playlist_id is duplicated:

In [8]:
```
duplicated_playlist_ids = (spotify_df_dict["playlist"].
 loc[spotify_df_dict["playlist"].duplicated(subset=['playlist_id'])]["playlist_id"])

spotify_df_dict["playlist"][spotify_df_dict["playlist"]["playlist_id"].
                            isin(duplicated_playlist_ids)].sort_values(by=["playlist_id"])
```

Out[8]:

	playlist_id	playlist_name	playlist_genre	playlist_subgenre
29945	25ButZrVb1Zj1MJioMs09D	EDM 2020 House & Dance	edm	pop edm
27216	25ButZrVb1Zj1MJioMs09D	EDM 2020 House & Dance	edm	electro house
30804	2CJsD3fcYJWcliEKnwmovU	TOP 50 GLOBAL 2020 UPDATED WEEKLY 🌎🎶 WORLDWIDE	edm	pop edm
23436	2CJsD3fcYJWcliEKnwmovU	TOP 50 GLOBAL 2020 UPDATED WEEKLY 🌎🎶 WORLDWIDE	r&b	hip pop
1067	37i9dQZF1DWTHM4kX49UKs	Ultimate Indie Presents... Best Indie Tracks o...	pop	dance pop
22829	37i9dQZF1DWTHM4kX49UKs	Ultimate Indie Presents... Best Indie Tracks o...	r&b	hip pop
10387	37i9dQZF1DX4OjfOteYnH8	Flow Selecto	rap	trap
19687	37i9dQZF1DX4OjfOteYnH8	Flow Selecto	latin	reggaeton
12900	3Ho3iO0iJykgEQNbjB2sic	Classic Rock 70s 80s 90s, Rock Classics - 70s ...	rock	classic rock
15155	3Ho3iO0iJykgEQNbjB2sic	Classic Rock 70s 80s 90s, Rock Classics - 70s ...	rock	hard rock
30196	3xMQTDLOIGvj3lWH5e5x6F	Charts 2020 🔥Top 2020🔥Hits 2020🔥Summer 2020🔥Po...	edm	pop edm
23099	3xMQTDLOIGvj3lWH5e5x6F	Charts 2020 🔥Top 2020🔥Hits 2020🔥Summer 2020🔥Po...	r&b	hip pop
19703	4JkkvMpVl4ISioqQjeAL0q	2020 Hits & 2019 Hits – Top Global Tracks 🔥🔥🔥	latin	latin hip hop
18295	4JkkvMpVl4ISioqQjeAL0q	2020 Hits & 2019 Hits – Top Global Tracks 🔥🔥🔥	latin	latin pop
23755	4JkkvMpVl4ISioqQjeAL0q	2020 Hits & 2019 Hits – Top Global Tracks 🔥🔥🔥	r&b	hip pop
27962	6KnQDwp0syvhfHOR4lWP7x	Fitness Workout Electro | House | Dance | Prog...	edm	electro house
31024	6KnQDwp0syvhfHOR4lWP7x	Fitness Workout Electro | House | Dance | Prog...	edm	progressive electro house

The issue seems to be that some playlists are listed under two subgenres. let's instead just create a new unique id for the playlist dataframe:

In [11]:
```
spotify_df_dict["playlist"]["playlist_id"] = list(range(0, spotify_df_dict["playlist"].shape[0]))
```

In [12]:
```
# Check playlist_uid is unique per row in playlist df
print("Is there a unique id per row?")
print(spotify_df_dict["playlist"].shape[0] == len(spotify_df_dict["playlist"]["playlist_uid"].unique()))

# Check playlist_uid has no NA values
print("Are there NA values?")
print(spotify_df_dict["playlist"]["playlist_id"].isnull().any())
```

```
Is there a unique id per row?
True
Are there NA values?
False
```

Check that there are no repeating columns

In [13]:
```
spotify_df_dict["track"].head()
```

Out[13]:

	track_id	track_name	track_artist	track_popularity	track_album_id	track_album_name	track_album_release_date
0	6f807x0ima9a1j3VPbc7VN	I Don't Care (with Justin Bieber) - Loud Luxur...	Ed Sheeran	66	2oCs0DGTsRO98Gh5ZSl2Cx	I Don't Care (with Justin Bieber) [Loud Luxury...	2019-06-14
1	0r7CVbZTWZgbTCYdfa2P31	Memories - Dillon Francis Remix	Maroon 5	67	63rPSO264uRjW1X5E6cWv6	Memories (Dillon Francis Remix)	2019-12-13
2	1z1Hg7Vb0AhHDiEmnDE79l	All the Time - Don Diablo Remix	Zara Larsson	70	1HoSmj2eLcsrR0vE9gThr4	All the Time (Don Diablo Remix)	2019-07-05
3	75FpbthrwQmzHlBJLuGdC7	Call You Mine - Keanu Silva Remix	The Chainsmokers	60	1nqYsOef1yKKuGOVchbsk6	Call You Mine - The Remixes	2019-07-19
4	1e8PAfcKUYoKkxPhrHqw4x	Someone You Loved - Future Humans Remix	Lewis Capaldi	69	7m7vv9wlQ4i0LFuJiE2zsQ	Someone You Loved (Future Humans Remix)	2019-03-05

In [14]:
```
spotify_df_dict["playlist"].head()
```

Out[14]:

	playlist_id	playlist_name	playlist_genre	playlist_subgenre	playlist_uid
0	0	Pop Remix	pop	dance pop	0
70	1	Dance Pop	pop	dance pop	1
167	2	Dance Room	pop	dance pop	2
223	3	Cardio	pop	dance pop	3
272	4	Dance Pop Hits	pop	dance pop	4

Repeating Columns

- For the track DataFrame, none of the columns seem to repeat
- For the playlist DataFrame there are no repeating groups of columns

Each cell contains a single value

- From our inspection of the DataFrames we see that each cell contains a single value; so, this requirement is satisfied.

Normalize data into Second Normal Form

Playlist table

In [15]:
```
spotify_df_dict["playlist"].columns
```

Out[15]:
```
Index(['playlist_id', 'playlist_name', 'playlist_genre', 'playlist_subgenre',
       'playlist_uid'],
      dtype='object')
```

Each column describes what the primary key identifies (a playlist), so 2NF is satisfied for this table.

Track table

In [16]:
```
spotify_df_dict["track"].columns
```

Out[16]:
```
Index(['track_id', 'track_name', 'track_artist', 'track_popularity',
       'track_album_id', 'track_album_name', 'track_album_release_date',
       'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness',
       'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo',
       'duration_ms'],
      dtype='object')
```

Here, we notice that some of the columns (specifically, track_album_id, track_album_name, track_album_release_date) relate to the album of the track, not the track itself (as the other columns do). So, let's move the album-related columns to their own table

In [17]:
```
album_columns = ["track_album_id",
                 "track_album_name",
                 "track_album_release_date"]

# Create new DataFrame of album information, dropping duplicates
spotify_df_dict["album"] = spotify_df_dict["track"][album_columns].drop_duplicates()

# Check that track_album_id is an appropriate primary key
print("Is there a unique id per row?")
print(spotify_df_dict["album"].shape[0] == len(spotify_df_dict["album"]["track_album_id"].unique()))

# Check playlist_uid has no NA values
print("Are there NA values?")
print(spotify_df_dict["album"]["track_album_id"].isnull().any())
```

```
Is there a unique id per row?
True
Are there NA values?
False
```

Now let's remove those columns from the track DataFrame and we should be all set with 2NF:

In [17]:
```
spotify_df_dict["track"].drop(album_columns, axis=1, inplace=True)

# Confirm that album columns were removed
spotify_df_dict["track"].columns
```

Out[17]:
```
Index(['track_id', 'track_name', 'track_artist', 'track_popularity',
       'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness',
       'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo',
       'duration_ms'],
      dtype='object')
```

Normalize data into Third Normal Form (3NF)

Album table

In [18]:
```
spotify_df_dict["album"].head()
```

Out[18]:

	track_album_id	track_album_name	track_album_release_date
0	2oCs0DGTsRO98Gh5ZSl2Cx	I Don't Care (with Justin Bieber) [Loud Luxury...	2019-06-14
1	63rPSO264uRjW1X5E6cWv6	Memories (Dillon Francis Remix)	2019-12-13
2	1HoSmj2eLcsrR0vE9gThr4	All the Time (Don Diablo Remix)	2019-07-05
3	1nqYsOef1yKKuGOVchbsk6	Call You Mine - The Remixes	2019-07-19
4	7m7vv9wlQ4i0LFuJiE2zsQ	Someone You Loved (Future Humans Remix)	2019-03-05

Each column depends on the album_id (or row number), so there are no 3NF violations.

Playlist table

In [19]:
```
spotify_df_dict["playlist"].head()
```

Out[19]:

	playlist_id	playlist_name	playlist_genre	playlist_subgenre	playlist_uid
0	37i9dQZF1DXcZDD7cfEKhW	Pop Remix	pop	dance pop	0
70	37i9dQZF1DWZQaaqNMbbXa	Dance Pop	pop	dance pop	1
167	37i9dQZF1DX2ENAPP1Tyed	Dance Room	pop	dance pop	2
223	37i9dQZF1DWSJHnPb1f0X3	Cardio	pop	dance pop	3
272	37i9dQZF1DX6pH08wMhkaI	Dance Pop Hits	pop	dance pop	4

- Here, playlist_subgenre violates 3NF, because it only depends on the playlist_id via the playlist_genre.
- So, let's split that out (with playlist_genre) into a separate genre table:

In [20]:
```
genre_columns = ["playlist_genre",
                 "playlist_subgenre"]

# Create new DataFrame of genre information, dropping duplicates
spotify_df_dict["genre"] = spotify_df_dict["playlist"][genre_columns].drop_duplicates()

# Check that playlist_subgenre is an appropriate primary key
print("Is there a unique id per row?")
print(spotify_df_dict["genre"].shape[0] == len(spotify_df_dict["genre"]["playlist_subgenre"].unique()))

# Check playlist_uid has no NA values
print("Are there NA values?")
print(spotify_df_dict["genre"]["playlist_subgenre"].isnull().any())
```

```
Is there a unique id per row?
True
Are there NA values?
False
```

Great. Lastly, let's remove playlist_genre and playlist_subgenre from the playlist table:

In [18]:
```
spotify_df_dict["playlist"].drop(["playlist_genre","playlist_subgenre"], axis=1, inplace=True)

# Confirm that album columns were removed
spotify_df_dict["playlist"].columns
```

Out[18]:
```
Index(['playlist_id', 'playlist_name', 'playlist_uid'], dtype='object')
```

Track table

In [19]:
```
spotify_df_dict["track"].head()
```

Out[19]:

	track_id	track_name	track_artist	track_popularity	track_album_id	track_album_name	track_album_release_date
0	6f807x0ima9a1j3VPbc7VN	I Don't Care (with Justin Bieber) - Loud Luxur...	Ed Sheeran	66	2oCs0DGTsRO98Gh5ZSl2Cx	I Don't Care (with Justin Bieber) [Loud Luxury...	2019-06-14
1	0r7CVbZTWZgbTCYdfa2P31	Memories - Dillon Francis Remix	Maroon 5	67	63rPSO264uRjW1X5E6cWv6	Memories (Dillon Francis Remix)	2019-12-13
2	1z1Hg7Vb0AhHDiEmnDE79l	All the Time - Don Diablo Remix	Zara Larsson	70	1HoSmj2eLcsrR0vE9gThr4	All the Time (Don Diablo Remix)	2019-07-05
3	75FpbthrwQmzHlBJLuGdC7	Call You Mine - Keanu Silva Remix	The Chainsmokers	60	1nqYsOef1yKKuGOVchbsk6	Call You Mine - The Remixes	2019-07-19
4	1e8PAfcKUYoKkxPhrHqw4x	Someone You Loved - Future Humans Remix	Lewis Capaldi	69	7m7vv9wlQ4i0LFuJiE2zsQ	Someone You Loved (Future Humans Remix)	2019-03-05

Each column depends on the track_id (or row number), so there are no 3NF violations.