**P2: Database Design, Conceptual Model**

**Group 13**

**Entities:**

1. **Tracks**: Contains information about each track, such as track name, release date (year, month, day), BPM (beats per minute), key, mode, danceability, valence, energy, acousticness, instrumentalness, liveness, and speechiness percentages.
2. **Artists**: Stores artist-related information. Each artist has a unique identifier and name. Tracks may be linked to one or more artists.
3. **Albums**: While the dataset does not explicitly include album names, if available, this entity could store album-related information and link to the tracks and artists involved.
4. **Playlists**: Information about Spotify playlists that include the track, such as playlist count and possibly the playlist names or IDs if available.
5. **Charts**: Details tracks' presence on Spotify charts, including the number of times it appeared on charts and stream counts.
6. **Artist\_Track\_Relationship**: A junction table to manage the many-to-many relationship between artists and tracks since a track can have multiple artists and an artist can have multiple tracks.
7. **Playlist\_Track\_Relationship**: Another junction table to manage the relationship between tracks and the playlists they appear in, useful if detailed playlist information is available or collected.
8. **Genres**: If genre data is available for each track or artist, this entity can store genre names. The relationship between tracks/artists and genres would likely be many-to-many, requiring additional junction tables.
9. **Charts\_Performance**: This entity could detail the performance of tracks on Spotify charts over time, including date-specific streams, chart positions, and duration on charts if this data can be captured or inferred.
10. **User\_Interaction**: This hypothetical entity could log user interactions with tracks, such as plays, likes, and adds to playlists, if such data were available, supporting detailed user behavior analytics.
11. **Release\_Information**: Specifically focuses on the release details of tracks, separating out the release date components for more complex temporal analyses.
12. **Track\_Features**: Focuses on the musical and audio features of tracks, such as BPM, key, mode, and various percentages (danceability, valence, energy, etc.), enabling analyses on music trends and preferences.
13. **Market\_Performance**: An entity to capture the market performance of tracks, including streams, chart appearances, and presence in playlists, possibly segmented by regions if such data is available or can be integrated.

**Entities with Attributes:**

1. **Tracks**
   * TrackID (Primary Key)
   * TrackName
   * ReleasedYear
   * ReleasedMonth
   * ReleasedDay
   * BPM
   * Key
   * Mode
   * DanceabilityPercent
   * ValencePercent
   * EnergyPercent
   * AcousticnessPercent
   * InstrumentalnessPercent
   * LivenessPercent
   * SpeechinessPercent
   * Streams
2. **Artists**
   * ArtistID (Primary Key)
   * ArtistName
3. **Albums** (Assuming album data can be added)
   * AlbumID (Primary Key)
   * AlbumName
   * ReleaseDate
   * ArtistID (Foreign Key)
4. **Playlists**
   * PlaylistID (Primary Key)
   * PlaylistName
   * InSpotifyPlaylistsCount
5. **Charts**
   * ChartID (Primary Key)
   * InSpotifyChartsCount
   * ChartDate (Assuming data on specific chart dates can be included)
6. **Artist\_Track\_Relationship**
   * ArtistID (Foreign Key)
   * TrackID (Foreign Key)
7. **Playlist\_Track\_Relationship**
   * PlaylistID (Foreign Key)
   * TrackID (Foreign Key)
8. **Genres** (Assuming genre data can be added)
   * GenreID (Primary Key)
   * GenreName
9. **Charts\_Performance**
   * PerformanceID (Primary Key)
   * TrackID (Foreign Key)
   * ChartID (Foreign Key)
   * StreamsOnDate
   * ChartPosition
10. **User\_Interaction** (Hypothetical, assuming user data can be added)
    * InteractionID (Primary Key)
    * UserID (Foreign Key)
    * TrackID (Foreign Key)
    * PlayCount
    * Like
    * AddedToPlaylist
11. **Release\_Information**
    * TrackID (Foreign Key)
    * ReleaseDate
    * ReleaseYear
    * ReleaseMonth
    * ReleaseDay
12. **Track\_Features**
    * TrackID (Foreign Key)
    * BPM
    * Key
    * Mode
    * DanceabilityPercent
    * ValencePercent
    * EnergyPercent
    * AcousticnessPercent
    * InstrumentalnessPercent
    * LivenessPercent
    * SpeechinessPercent
13. **Market\_Performance**
    * TrackID (Foreign Key)
    * Streams
    * InSpotifyPlaylistsCount
    * InSpotifyChartsCount

**Entities and Relationships:**

1. **Tracks to Artists** (Many-to-Many): A track can have multiple artists, and an artist can have multiple tracks. This relationship is managed through the **Artist\_Track\_Relationship** junction table.
2. **Tracks to Albums** (Many-to-One): Each track belongs to one album, but an album can contain multiple tracks. This assumes album data is available and included in your schema.
3. **Artists to Albums** (Many-to-Many): An artist can release multiple albums, and an album can feature multiple artists. If albums are tracked, a junction table might be necessary unless albums are uniquely linked to single artists.
4. **Tracks to Playlists** (Many-to-Many): A track can be included in multiple playlists, and a playlist can contain multiple tracks. This relationship is managed through the **Playlist\_Track\_Relationship** junction table.
5. **Tracks to Genres** (Many-to-Many): A track can belong to multiple genres, and a genre can include many tracks. This relationship would require a junction table if genre data is available and included.
6. **Artists to Genres** (Many-to-Many): An artist can produce works in multiple genres, and a genre can encompass works by many artists. This relationship also would likely require a junction table for proper management.
7. **Tracks to Charts\_Performance** (One-to-Many): Each track can have multiple chart performance entries (reflecting its performance over time), but each chart performance entry is linked to one track.
8. **Tracks to Market\_Performance** (One-to-One): Each track has one market performance entry that summarizes its overall market performance, including streams and presence in playlists and charts.
9. **Tracks to Track\_Features** (One-to-One): Each track has a set of features (like BPM, key, mode, etc.) that are unique to it.
10. **Tracks to Release\_Information** (One-to-One): Each track has one release information entry detailing its release date.
11. **User\_Interaction to Tracks** (Many-to-One): Assuming user interaction data is available, each interaction (like play, like, add to playlist) is associated with one track, but a track can have many interactions.