Description of Entities, Relationships, and Attributes

Entities:

- User: Entity that holds user information in order to perform songs
- Time: Entity that only records the time when a song is performed
- Queue: Entity that is a supertype to handle the free and paid queues
- FQ: Subtype of Queue via "GD", this is the free queue i.e., first come first served
- PQ: Subtype of Queue via "GD", this is the paid queue where whoever pays the most has their song at the front.
- KaraokeFile: Main entity that holds all information of the file/version via multiple relationships.
- Title: Entity that stores data relating to the title of the karaoke files
- Artist: Entity that stores data relating to the artist of karaoke files
- Contributor: Entity that stores data relating to those who worked on the karaoke files in collaboration with the artist.

Relationships:

- KaraokeFile (1,M) has → (1,1) Artist: There are many karaoke files available, and each file will have one artist, under the assumption that features are technically contributors via the "vocals" position.
- KaraokeFile (1,M) has → (1,1) Title: There are many karaoke files available, and each one will have one title assigned to it.
- Contributor (1,M) cont. → (1,M) KaraokeFile: Many contributors can contribute to many karaoke files via a multitude of positions.
- User (1,M) Performs → (1,M) KaraokeFile: Many users can perform many karaoke files throughout the duration of the time it is open.
- User (1,M) Performs → (1,M) Time: Many users can perform at many different times, although this relationship does not hold vital information.
- User (1,M) Performs → (1,1) Queue: Many users can perform, but they can only enter one queue at a time. This is demonstrated via Queue being a supertype connected to the two queues, "FQ" and "PQ", via "GD". The two queues are subtypes and any one user can be in one or the other, but not at the same time in order to let other users have a fair chance.

Attributes:

- User Attributes:
  - Name: Identifying name of the User
  - UserID: Primary key and unique ID assigned to each user
- PQ Attributes:
  - PQID: Primary key and ID for user entering queue
  - Playing: Either true or false
  - Price: How much money was paid
  - Time: Time entered queue
- FQ Attributes:
  - FQID: Primary key and ID for user entering queue
  - Playing: Either true or false

    - Time: Time entered queue
- KaraokeFile Attributes:
    - FileID: Primary key and unique ID assigned to each file
    - Version: version of the file because some may have duet versions
- Title Attributes:
    - Name: Name of the title
    - TitleID: Primary key and unique identifier for each title
- Artist Attributes:
    - Name: Name of the artist which can be real or stage name
    - ArtistID: Primary key and unique identifier for each artist
- Contributor Attributes:
    - Name: Name of contributor which can be either real or stage name
    - ConID: Primary key and unique identifier for each contributor
- Contributes Attributes:
    - Position: This relational attribute is used to show the position of the contributor in the karaoke file.

Relational Schema

The schema is listed below. Underlined indicates a primary key and italics indicates a foreign key. Home relations will be indented below each relation if applicable. (Omissions: Time is not included in the relational schema because it's a date-like entity. Queue's subtypes are included but not Queue itself because its strong subtype entities serve the purpose.)

- **Artist** (Name, ArtistID)
- **Title** (Name, TitleID)
- **Contributor** (Name, ConID)
- **User** (Name, UserID)
- **FQ** (Time, Playing, FQID)
- **PQ** (Time, Price, Playing, PQID)
- **Contributes** (*FileID*, *ConID*, Position)
    - Home relation of "*FileID*": FileID is the primary key for KaraokeFile, which makes its use here as a foreign key, thus needing a home relation.
    - Home relation of "*ConID*": ConID is the primary key for Contributor, which makes its use here as a foreign key, thus needing a home relation.
- **KaraokeFile** (FileID, Version, *TitleID*, *ArtistID*, *ConID*)
    - Home relation of "*TitleID*": TitleID is the primary key for Title, which makes its use here as a foreign key, thus needing a home relation.
    - Home relation of "*ArtistID*": ArtistID is they primary key for Artist, which makes its use here as a foreign key, thus needing a home relation.
    - Home relation of "*ConID*": ConID is the primary key for Contributor, which makes its use here as a foreign key, thus needing a home relation.