CSCI 466 Relational Schema

Song (ID, CoverArt, Title, Genre, Duration)

Artist (ID, Name, Band)

Role (ID, Type)

KaraokeFile (ID, FileLocation)

Queue (ID, AmountPaid, Status)

Person (ID, Email, First Name, Last Name, AddressLine1, AddressLine2)

Client (ID, Email, First Name, Last Name, AddressLine1, AddressLine2)

DJ (ID, Email, First Name, Last Name, AddressLine1, AddressLine2)

Priority (ID, AmountPayed, Status)

FFA (ID, AmountPayed, Status)

AssociatedWith (Artist_Identifier^t, KF_Identifier^t, Song_Identifier^t)

Contributes (Song_Identifier^t, Role_Identifier^t, Artist_Identifier^t, Date)

Enqueues (KF_Identifier^t, Person_Identifier^t, Queue_Identifier^t, Date, Price, Dequeued)

Description of foreign keys and their home relations.

Foreign Key ^{t}	Home relation
Artist_Identifier^t	Artist
KF_Identifier^t	KaraokeFile
Song_Identifier^t	Song
Role_Identifier^t	Role
Person_Identifier^t	Person
Queue_Identifier^t	Queue

Detailed description of each Relation and it's attributes

Song
The song relation is used to hold data such as cover art, title, genre, and duration relating to the songs in the database. The primary key for this relation is ID, which is a surrogate key that is generated to uniquely identify each song.

Artist
The artist relation is used to hold data such as Name, association, and band relating to the artists in the database. The primary key for this relation is ID, which is a surrogate key that is generated to uniquely identify each artist.

Role
The role relation is used to hold data such as the type, relating to the role in the database. The primary key for this relation is ID, which is a surrogate key that is generated to uniquely identify each role.

KaraokeFile
The karaoke file relation is used to hold data such as file location relating to the song files in the database. The primary key for this relation is ID, which is a surrogate key that is generated to uniquely identify each song file.

Queue
The queue relation is the supertype from witch the priority and FFA relations inherit using generalized disjoint is-a inheritance. The primary key for this relation is ID, which is a surrogate key that is generated to uniquely identify each queue. The AmountPayed tracks the amount a client has paid to get onto the priority queue. The Status flag is used to determine if a client has paid or not.

Person
The person relation is used to hold data such as email, first name, last name, address line 1, and address line 2 relating to the people who use this application. This relation is the supertype that the client and DJ relations inherit from using specialized overlapping is-a inheritance. The primary key for this relation is ID, which is a surrogate key that is generated to uniquely identify each person.

Client
The person relation is used to hold data such as email, first name, last name, address line 1, and address line 2 relating to the clients who use this application. The primary key for this relation is ID, which is a surrogate key that is generated to uniquely identify each client. This relation is a subtype of the person relation.

DJ
The DJ relation is used to hold data such as email, first name, last name, address line 1, and address line 2 relating to the DJs who use this application. The primary key for this relation is ID, which is a surrogate key that is generated to uniquely identify each DJ. This relation is a subtype of the person relation.

Priority
The priority relation is a type of queue used to hold data such as the ID relating to the clients who have paid money to use the application. The primary key for this relation is ID, which is a surrogate key that is generated to uniquely identify each paying client.

FFA
The FFA relation is a type of queue used to hold data such as the ID relating to the clients who have not paid money to use the application. The primary key for this relation is ID, which is a surrogate key that is generated to uniquely identify each non-paying client.

AssociatedWith
The AssociatedWith relation is used to hold data such as which artists perform a given song in the database. The primary key for this relation is made up of Artist_Identifier^{t} and, KF_Identifier^{t}. Song_Identifier^{t} Artist_Identifier^{t} and, KF_Identifier^{t} are all foreign keys, and their home relations are listed above. This relation is used to associate the Artist, KaraokeFile, and Song relations together.

Contributes
The Contributes relation is used to hold data such as date, relating to when a contribution was made to a song in the database. The primary key for this relation is made up of Song_Identifier^{t}, Role_Identifier^{t}, and Artist_Identifier^{t}. This relation is used to associate an Artist, Song and, Role to find out when and what an artist contributed to a song.

Enqueues
The enqueues relation is used to hold data such as date, price, and dequeued relating to when a person chooses a karaoke file and subsequent queue. The primary key for this relation is made up of KF_Identifier^{t}, Person_Identifier^{t}, Queue_Identifier^{t}. This relation is used to associate a person, karaoke file, and queue to find out who is singing what and if they are a paying client or not. This relation also stores data on who has already been up to sing.