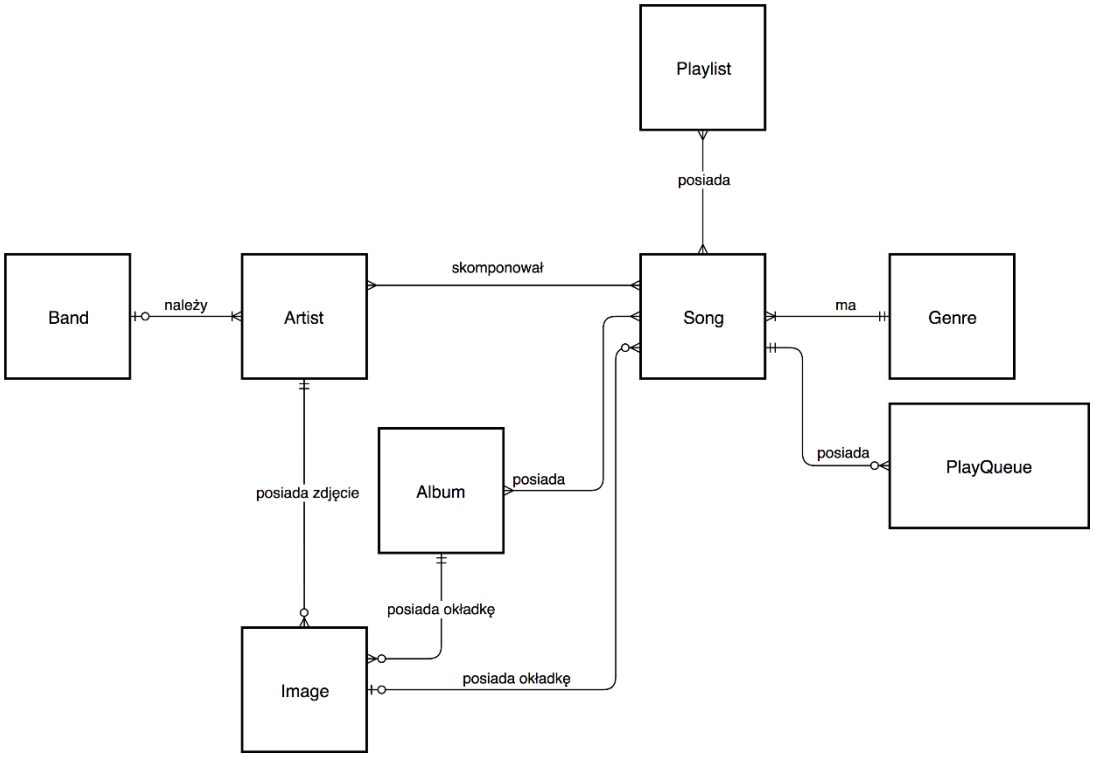
Wydział Informatyki Politechniki Białostockiej Pracowania specjalistyczna Systemy baz danych	Data oddania: 29.11.2018r.
SPRAWOZDANIE Z PROJEKTU (Etap 1 i 2) Temat: Odtwarzacz muzyczny Grupa Ps 5 Autorzy: 1. Adam Bajguz 2. Magdalena Kalisz 3. Michał Kierzkowski	Prowadzący: mgr inż. Joanna Gościk Ocena:

1 OPIS PROJEKTU

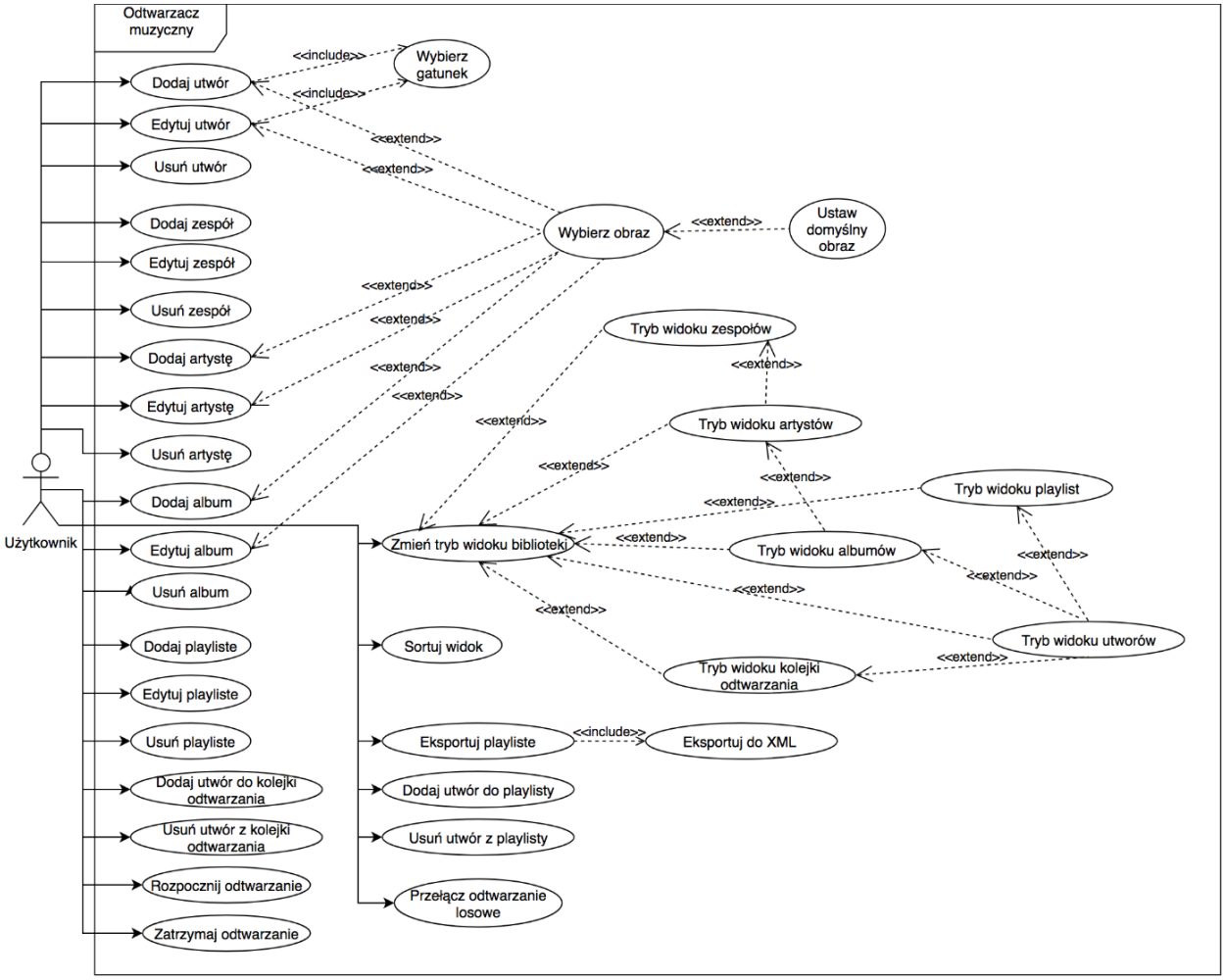
Odtwarzacz muzyczny na komputer PC z systemem Windows dla jednego użytkownika działający jako aplikacja Universal Windows Platform:

- obsługa podstawowych formatów plików dźwiękowych (przynajmniej WAV i MP3);
- playlisty odtwarzanych plików z możliwością edycji (dodawanie, usuwanie, zmiana kolejności);
- zapis playlist i ich eksport (do formatu XML i/lub JSON);
- biblioteka utworów z prezentacją w widokach względem nazwy artysty, jego albumów i ścieżek w albumie;
- możliwość sortowania widoków biblioteki przynajmniej po nazwie, roku wydania, długości ścieżki;
- grupowanie utworów w albumy (jeden utwór może znajdować się w kilku albumach);
- każdy utwór ma mieć przypisany dokładnie jeden gatunek (lista gatunków jest ustalana przez użytkownika, tzn. użytkownik dodaje, usuwa i edytuje dostępne gatunki);
- każdy utwór ma mieć możliwość dodania własnej grafiki, jeśli jej nie ma wyświetlana jest domyślna grafika zapisana w aplikacji lub wyświetlana jest okładka albumu o ile istnieje;
- każdy utwór w albumie ma przypisany numer ścieżki;
- przypisywanie albumu do artysty (artysta może posiadać wiele albumów);
- przypisywanie artysty do zespołu (artysta może być tylko w jednym zespole);
- album, utwór oraz artysta mogą posiadać dokładnie jedno zdjęcie/okładkę;
- oprócz playlist powinna istnieć również kolejka odtwarzania zawierająca wszystkie utwory do odtworzenia;
- użytkownik ma mieć możliwość dodania albumu/playlisty lub pojedynczego utworu do kolejki odtwarzania.

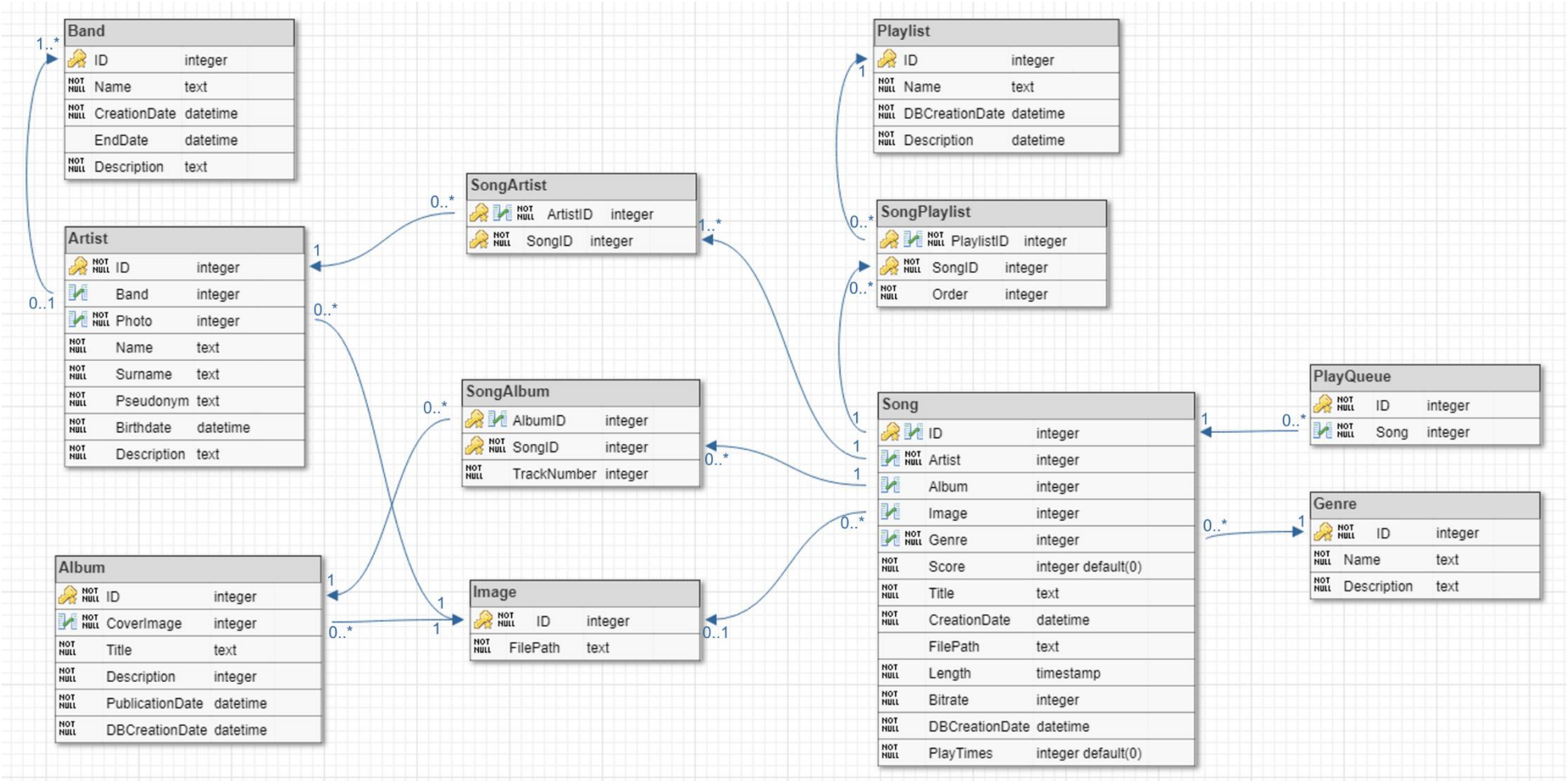
2 MODEL KONCEPCYJNY PROJEKTU BAZY DANYCH (DIAGRAM ER)



3 DIAGRAM PRZYPADKÓW UŻYCIA



4 PROJEKT RELACYJNEJ BAZY DANYCH



5 OPIS TABEL I PÓL

Song – tabela przechowująca dane o utworze

NAZWA POLA	TYP	OPIS
ID	integer	Unikalny identyfikator każdego utworu
Artist	integer	Klucz obcy powiązany z tabelą SongArtist
Album	integer	Klucz obcy powiązany z tabelą SongAlbum
Image	integer	Klucz obcy powiązany z tabelą Image; przechowuje grafikę utworu
Genre	Integer	Klucz obcy powiązany z tabelą Genre; przechowuje gatunek utworu
Score	integer	Wartość liczbowa reprezentująca ocenę utworu przez użytkownika
Title	text	Tytuł utworu
CreationDate	datetime	Data powstania/wydania utworu
FilePath	text	Ścieżka do pliku z utworem na dysku użytkownika
Length	timestamp	Długość utworu
Bitrate	integer	Wartość liczbowa reprezentująca bitrate utworu
DBCreationDate	datetime	Data utworzenia rekordu utworu w bazie
PlayTimes	integer	Liczba odtworzenia utworu

Genre – tabela przechowująca dane o gatunku muzycznym zdefiniowanym przez użytkownika

NAZWA POLA	TYP	OPIS
ID	integer	Unikalny identyfikator każdego gatunku
Name	text	Nazwa gatunku
Description	text	Opis tekstowy gatunku

PlayQueue – tabela przechowująca klucze utworów które mają zostać odtworzone, stanowi ona kolejkę odtwarzania utworów

NAZWA POLA	TYP	OPIS
ID	integer	Unikalny identyfikator każdego elementu kolejki odtwarzania
Song	integer	Klucz obcy powiązany z tabelą Song. Reprezentuje utwór do odtworzenia

SongPlaylist – tabela wiążąca tabelę Song i tabelę Playlist w celu zapewniania relacji wiele do wielu oraz przechowania informacji o kolejności utworów w Playliście

NAZWA POLA	TYP	OPIS
PlaylistID	integer	Klucz obcy powiązany z tabelą Playlist; element klucza złożonego
SongID	integer	Klucz obcy powiązany z tabelą Song; element klucza złożonego
Order	integer	Numer porządkowy utworu na playliście służący do wyznaczania kolejności utworów w playliście (sortowania utworów)

Playlist – tabela przechowująca dane o playliście

NAZWA POLA	TYP	OPIS
ID	integer	Unikalny identyfikator każdej playlisty
Name	text	Nazwa playlisty
DBCreationDate	datetime	Data utworzenia rekordu playlisty w bazie
Description	text	Opis tekstowy playlisty

Image – tabela przechowująca dane o grafikach używanych w aplikacji od okładek, zdjęć artystów

NAZWA POLA	TYP	OPIS
ID	integer	Unikalny identyfikator każdego obrazu
FilePath	text	Ścieżka do pliku z obrazem na dysku użytkownika

SongAlbum – tabela wiążąca tabelę Song i tabelę Album w celu zapewniania relacji wiele do wielu oraz przechowania informacji o numerze ścieżki utworu w Albumie

NAZWA POLA	TYP	OPIS
AlbumID	integer	Klucz obcy powiązany z tabelą Album; element klucza złożonego
SongID	integer	Klucz obcy powiązany z tabelą Song; element klucza złożonego
TrackNumber	integer	Numer ścieżki w albumie

Album – tabela przechowująca dane albumów

NAZWA POLA	TYP	OPIS
ID	integer	Unikalny identyfikator każdego albumu
CoverImage	integer	Klucz obcy powiązany z tabelą Image; przechowuje okładkę albumu
Title	text	Nazwa albumu
Description	integer	Opis tekstowy albumu
PublicationDate	datetime	Data powstania/wydania albumu
DBCreationDate	datetime	Data utworzenia rekordu albumu w bazie

Song Artist – tabela wiążąca tabelę Song i tabelę Artist w celu zapewniania relacji wiele do wielu

NAZWA POLA	TYP	OPIS
ArtistID	integer	Klucz obcy powiązany z tabelą Artist
SongID	integer	Klucz obcy powiązany z tabelą Song

Artist – tabele przechowująca dane artystów

NAZWA POLA	TYP	OPIS
ID	integer	Unikalny identyfikator każdego artysty
Band	integer	Klucz obcy powiązany z tabelą Band. Reprezentuje zespół do którego należy artysta
Image	integer	Klucz obcy powiązany z tabelą Image. Reprezentuje zdjęcie artysty
Name	text	Imię artysty
Surname	text	Nazwisko artysty
Pseudonym	text	Pseudonim artystyczny artysty
Description	text	Opis tekstowy artysty, np. jego bibliografia

Zespol – tabela przechowująca dane zespołów

NAZWA POLA	TYP	OPIS
ID	integer	Unikalny identyfikator każdego zespołu
Name	text	Nazwa zespołu
CreationData	datetime	Data powstania/założenia zespołu
EndDate	datetime	Data rozwiązania zespołu
Description	text	Opis tekstowy zespołu

6 WYBÓR SERWERA BAZODANOWEGO I IMPLEMENTACJA BAZY DANYCH

Do stworzenia implementacji wykorzystamy Microsoft SQL Server.

Skrypt tworzący bazę:

```
CREATE TABLE [Album] (
        ID integer NOT NULL,
        CoverImage integer NOT NULL,
        Title text NOT NULL,
        Description integer NOT NULL,
        PublicationDate datetime NOT NULL,
        DBCreationDate datetime NOT NULL,
  CONSTRAINT [PK_ALBUM] PRIMARY KEY CLUSTERED
  (
  [ID] ASC
  ) WITH (IGNORE_DUP_KEY = OFF)

)
GO
CREATE TABLE [Song] (
        ID integer NOT NULL UNIQUE,
        Artist integer NOT NULL,
        Album integer,
        Image integer,
        Genre integer NOT NULL,
        Score integer NOT NULL DEFAULT '0',
        Title text NOT NULL,
        CreationDate datetime NOT NULL,
        FilePath text,
        Length timestamp NOT NULL,
        Bitrate integer NOT NULL,
        DBCreationDate datetime NOT NULL,
        PlayTimes integer NOT NULL DEFAULT '0',
  CONSTRAINT [PK_SONG] PRIMARY KEY CLUSTERED
  (
  [ID] ASC
  ) WITH (IGNORE_DUP_KEY = OFF)

)
GO
```

```
CREATE TABLE [Playlist] (
        ID integer NOT NULL,
        Name text NOT NULL,
        DBCreationDate datetime NOT NULL,
        Description datetime NOT NULL,
  CONSTRAINT [PK_PLAYLIST] PRIMARY KEY CLUSTERED
  (
  [ID] ASC
  ) WITH (IGNORE_DUP_KEY = OFF)

)
GO
CREATE TABLE [Artist] (
        ID integer NOT NULL,
        Band integer,
        Photo integer NOT NULL,
        Name text NOT NULL,
        Surname text NOT NULL,
        Pseudonym text NOT NULL,
        Birthdate datetime NOT NULL,
        Description text NOT NULL,
  CONSTRAINT [PK_ARTIST] PRIMARY KEY CLUSTERED
  (
  [ID] ASC
  ) WITH (IGNORE_DUP_KEY = OFF)

)
GO
CREATE TABLE [SongAlbum] (
        AlbumID integer NOT NULL,
        SongID integer NOT NULL,
        TrackNumber integer NOT NULL,
  CONSTRAINT [PK_SONGALBUM] PRIMARY KEY CLUSTERED
  (
  [AlbumID] ASC
  ) WITH (IGNORE_DUP_KEY = OFF)

)
GO
CREATE TABLE [SongArtist] (
        ArtistID integer NOT NULL,
        SongID integer NOT NULL,
  CONSTRAINT [PK_SONGARTIST] PRIMARY KEY CLUSTERED
  (
  [ArtistID] ASC
  ) WITH (IGNORE_DUP_KEY = OFF)

)
GO
CREATE TABLE [Image] (
        ID integer NOT NULL,
        FilePath text NOT NULL,
  CONSTRAINT [PK_IMAGE] PRIMARY KEY CLUSTERED
  (
  [ID] ASC
  ) WITH (IGNORE_DUP_KEY = OFF)

)
GO
CREATE TABLE [SongPlaylist] (
        PlaylistID integer NOT NULL,
        SongID integer NOT NULL,
        Order integer NOT NULL,
  CONSTRAINT [PK_SONGPLAYLIST] PRIMARY KEY CLUSTERED
  (
  [PlaylistID] ASC
  ) WITH (IGNORE_DUP_KEY = OFF)

)
GO
CREATE TABLE [Band] (
        ID integer NOT NULL,
        Name text NOT NULL,
        CreationDate datetime NOT NULL,
        EndDate datetime,
        Description text NOT NULL,
  CONSTRAINT [PK_BAND] PRIMARY KEY CLUSTERED
  (
  [ID] ASC
  ) WITH (IGNORE_DUP_KEY = OFF)
```

```
)
GO
CREATE TABLE [Genre] (
        ID integer NOT NULL,
        Name text NOT NULL,
        Description text NOT NULL,
  CONSTRAINT [PK_GENRE] PRIMARY KEY CLUSTERED
  (
  [ID] ASC
  ) WITH (IGNORE_DUP_KEY = OFF)

)
GO
CREATE TABLE [PlayQueue] (
        ID integer NOT NULL,
        Song integer NOT NULL,
  CONSTRAINT [PK_PLAYQUEUE] PRIMARY KEY CLUSTERED
  (
  [ID] ASC
  ) WITH (IGNORE_DUP_KEY = OFF)

)
GO
ALTER TABLE [Album] WITH CHECK ADD CONSTRAINT [Album_fk0] FOREIGN KEY ([CoverImage]) REFERENCES
[Image]([ID])
ON UPDATE CASCADE
GO
ALTER TABLE [Album] CHECK CONSTRAINT [Album_fk0]
GO

ALTER TABLE [Song] WITH CHECK ADD CONSTRAINT [Song_fk0] FOREIGN KEY ([ID]) REFERENCES
[SongPlaylist]([SongID])
ON UPDATE CASCADE
GO
ALTER TABLE [Song] CHECK CONSTRAINT [Song_fk0]
GO
ALTER TABLE [Song] WITH CHECK ADD CONSTRAINT [Song_fk1] FOREIGN KEY ([Artist]) REFERENCES
[SongArtist]([SongID])
ON UPDATE CASCADE
GO
ALTER TABLE [Song] CHECK CONSTRAINT [Song_fk1]
GO
ALTER TABLE [Song] WITH CHECK ADD CONSTRAINT [Song_fk2] FOREIGN KEY ([Album]) REFERENCES
[SongAlbum]([SongID])
ON UPDATE CASCADE
GO
ALTER TABLE [Song] CHECK CONSTRAINT [Song_fk2]
GO
ALTER TABLE [Song] WITH CHECK ADD CONSTRAINT [Song_fk3] FOREIGN KEY ([Image]) REFERENCES
[Image]([ID])
ON UPDATE CASCADE
GO
ALTER TABLE [Song] CHECK CONSTRAINT [Song_fk3]
GO
ALTER TABLE [Song] WITH CHECK ADD CONSTRAINT [Song_fk4] FOREIGN KEY ([Genre]) REFERENCES
[Genre]([ID])
ON UPDATE CASCADE
GO
ALTER TABLE [Song] CHECK CONSTRAINT [Song_fk4]
GO


ALTER TABLE [Artist] WITH CHECK ADD CONSTRAINT [Artist_fk0] FOREIGN KEY ([Band]) REFERENCES
[Band]([ID])
ON UPDATE CASCADE
GO
ALTER TABLE [Artist] CHECK CONSTRAINT [Artist_fk0]
GO
ALTER TABLE [Artist] WITH CHECK ADD CONSTRAINT [Artist_fk1] FOREIGN KEY ([Photo]) REFERENCES
[Image]([ID])
ON UPDATE CASCADE
GO
ALTER TABLE [Artist] CHECK CONSTRAINT [Artist_fk1]
GO

ALTER TABLE [SongAlbum] WITH CHECK ADD CONSTRAINT [SongAlbum_fk0] FOREIGN KEY ([AlbumID]) REFERENCES
[Album]([ID])
ON UPDATE CASCADE
GO
ALTER TABLE [SongAlbum] CHECK CONSTRAINT [SongAlbum_fk0]
GO
```

```
ALTER TABLE [SongArtist] WITH CHECK ADD CONSTRAINT [SongArtist_fk0] FOREIGN KEY ([ArtistID])
REFERENCES [Artist]([ID])
ON UPDATE CASCADE
GO
ALTER TABLE [SongArtist] CHECK CONSTRAINT [SongArtist_fk0]
GO


ALTER TABLE [SongPlaylist] WITH CHECK ADD CONSTRAINT [SongPlaylist_fk0] FOREIGN KEY ([PlaylistID])
REFERENCES [Playlist]([ID])
ON UPDATE CASCADE
GO
ALTER TABLE [SongPlaylist] CHECK CONSTRAINT [SongPlaylist_fk0]
GO



ALTER TABLE [PlayQueue] WITH CHECK ADD CONSTRAINT [PlayQueue_fk0] FOREIGN KEY ([Song]) REFERENCES
[Song]([ID])
ON UPDATE CASCADE
GO
ALTER TABLE [PlayQueue] CHECK CONSTRAINT [PlayQueue_fk0]
GO

////////////////////////////////////////////////////////////////////////////////
```