

DDL COMMANDS FOR MUSIC DATABASE

QUERIES

```
1
2 create table if not exists user_types(
3     type_id int primary key,
4     user_role varchar(15),
5     constraint user_role_key unique(user_role)
6 );
7
8 create table if not exists users(
9     user_id int primary key,
10    user_type int,
11    first_name varchar(30),
12    last_name varchar(30),
13    gender varchar(10),
14    birth_date date,
15    email varchar(80),
16    password varchar(125),
17    image_id int,
18    last_login date,
19    active_status bool,
20    constraint email_key unique(email),
21    constraint fk_user_type foreign key (user_type) references user_types(type_id)
22 );
23
```

```
24 create table if not exists genre(  
25     genre_id int primary key,  
26     genre_name varchar(15),  
27     constraint genre_name_key unique (genre_name)  
28 );  
29  
30 create table if not exists image(  
31     image_id int primary key,  
32     image_location varchar(250),  
33     image_desc varchar (100)  
34 );  
35  
36 create table if not exists lyrics(  
37     lyrics_id int primary key,  
38     lyrics_description varchar(1000),  
39     creator_name varchar(50)  
40 );  
41  
42 create table if not exists playlist(  
43     playlist_id int primary key,  
44     user_id int,  
45     playlist_name varchar(15),  
46     constraint fk_playlist_user_id foreign key (user_id) references users(user_id)  
47 );  
48  
49 create table if not exists artist(  
50     artist_id int primary key,  
51     artist_name varchar(50),  
52     total_songs int,  
53     total_albums int,  
54     image_id int,  
55     constraint artist_name_key unique(artist_name),  
56     constraint fk_artist_image_id foreign key (image_id) references image(image_id)  
57 );  
58
```

```

59 create table if not exists album(
60     album_id int primary key,
61     album_name varchar(20),
62     release_date date,
63     album_length float,
64     total_tracks int,
65     artist_id int,
66     genre_id int,
67     image_id int,
68     download_pemr bool,
69     constraint fk_album_genre_id foreign key (genre_id) references genre(genre_id),
70     constraint fk_album_artist_id foreign key (artist_id) references artist(artist_id),
71     constraint fk_album_image_id foreign key (image_id) references image(image_id)
72 );
73
74
75 create table if not exists songs(
76     song_id int primary key,
77     title varchar(20),
78     lenth float,
79     album_id int,
80     genre_id int,
81     lyrics_id int,
82     playlist_id int,
83     download_perm bool,
84     constraint fk_song_genre_id foreign key (genre_id) references genre(genre_id),
85     constraint fk_song_artist_id foreign key (album_id) references album(album_id),
86     constraint fk_songs_lyrics_id foreign key (lyrics_id) references lyrics(lyrics_id),
87     constraint fk_songs_playlist_id foreign key (playlist_id) references playlist(playlist_id)
88 );
89

```

OUTPUTS

```

Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (11.8)
WARNING: Console code page (437) differs from Windows code page (1252)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.
Type "help" for help.

```

```

postgres=# \c music_db
You are now connected to database "music_db" as user "postgres".
music_db=# \d user_types

```

Table "public.user_types"				
Column	Type	Collation	Nullable	Default
type_id	integer		not null	
user_role	character varying(15)			

```

Indexes:
    "user_types_pkey" PRIMARY KEY, btree (type_id)
    "user_role_key" UNIQUE CONSTRAINT, btree (user_role)
Referenced by:
    TABLE "users" CONSTRAINT "fk_user_type" FOREIGN KEY (user_type) REFERENCES user_types(type_id)

```

```
music_db=#
```

```
music_db=# \d users
```

Table "public.users"				
Column	Type	Collation	Nullable	Default
user_id	integer		not null	
user_type	integer			
first_name	character varying(30)			
last_name	character varying(30)			
gender	character varying(10)			
birth_date	date			
email	character varying(80)			
password	character varying(125)			
image_id	integer			
last_login	date			
active_status	boolean			

```

Indexes:
    "users_pkey" PRIMARY KEY, btree (user_id)
    "email_key" UNIQUE CONSTRAINT, btree (email)
Foreign-key constraints:
    "fk_user_type" FOREIGN KEY (user_type) REFERENCES user_types(type_id)
Referenced by:
    TABLE "playlist" CONSTRAINT "fk_playlist_user_id" FOREIGN KEY (user_id) REFERENCES users(user_id)

```

```
music_db=#
```

music_db=# \d genre

Table "public.genre"				
Column	Type	Collation	Nullable	Default
genre_id	integer		not null	
genre_name	character varying(15)			

Indexes:

"genre_pkey" PRIMARY KEY, btree (genre_id)

"genre_name_key" UNIQUE CONSTRAINT, btree (genre_name)

Referenced by:

TABLE "album" CONSTRAINT "fk_album_genre_id" FOREIGN KEY (genre_id) REFERENCES genre(genre_id)

TABLE "songs" CONSTRAINT "fk_song_genre_id" FOREIGN KEY (genre_id) REFERENCES genre(genre_id)

music_db=#

music_db=# \d image

Table "public.image"				
Column	Type	Collation	Nullable	Default
image_id	integer		not null	
image_location	character varying(250)			
image_desc	character varying(100)			

Indexes:

"image_pkey" PRIMARY KEY, btree (image_id)

Referenced by:

TABLE "album" CONSTRAINT "fk_album_image_id" FOREIGN KEY (image_id) REFERENCES image(image_id)

TABLE "artist" CONSTRAINT "fk_artist_image_id" FOREIGN KEY (image_id) REFERENCES image(image_id)

music_db=#

music_db=# \d lyrics

Table "public.lyrics"				
Column	Type	Collation	Nullable	Default
lyrics_id	integer		not null	
lyrics_description	character varying(1000)			
creator_name	character varying(50)			

Indexes:

"lyrics_pkey" PRIMARY KEY, btree (lyrics_id)

Referenced by:

TABLE "songs" CONSTRAINT "fk_songs_lyrics_id" FOREIGN KEY (lyrics_id) REFERENCES lyrics(lyrics_id)

music_db=#

music_db=# \d playlist

Table "public.playlist"				
Column	Type	Collation	Nullable	Default
playlist_id	integer		not null	
user_id	integer			
playlist_name	character varying(15)			

Indexes:

"playlist_pkey" PRIMARY KEY, btree (playlist_id)

Foreign-key constraints:

"fk_playlist_user_id" FOREIGN KEY (user_id) REFERENCES users(user_id)

Referenced by:

TABLE "songs" CONSTRAINT "fk_songs_playlist_id" FOREIGN KEY (playlist_id) REFERENCES playlist(playlist_id)

music_db=#

music_db=# \d artist

Table "public.artist"				
Column	Type	Collation	Nullable	Default
artist_id	integer		not null	
artist_name	character varying(50)			
total_songs	integer			
total_albums	integer			
image_id	integer			

Indexes:

"artist_pkey" PRIMARY KEY, btree (artist_id)

"artist_name_key" UNIQUE CONSTRAINT, btree (artist_name)

Foreign-key constraints:

"fk_artist_image_id" FOREIGN KEY (image_id) REFERENCES image(image_id)

Referenced by:

TABLE "album" CONSTRAINT "fk_album_artist_id" FOREIGN KEY (artist_id) REFERENCES artist(artist_id)

music_db=#

music_db=# \d album

Table "public.album"				
Column	Type	Collation	Nullable	Default
album_id	integer		not null	
album_name	character varying(20)			
release_date	date			
album_length	double precision			
total_tracks	integer			
artist_id	integer			
genre_id	integer			
image_id	integer			
download_pemr	boolean			

Indexes:

"album_pkey" PRIMARY KEY, btree (album_id)

Foreign-key constraints:

"fk_album_artist_id" FOREIGN KEY (artist_id) REFERENCES artist(artist_id)

"fk_album_genre_id" FOREIGN KEY (genre_id) REFERENCES genre(genre_id)

"fk_album_image_id" FOREIGN KEY (image_id) REFERENCES image(image_id)

Referenced by:

TABLE "songs" CONSTRAINT "fk_song_artist_id" FOREIGN KEY (album_id) REFERENCES album(album_id)

music_db=#

music_db=# \d songs

Table "public.songs"				
Column	Type	Collation	Nullable	Default
song_id	integer		not null	
title	character varying(20)			
lenth	double precision			
album_id	integer			
genre_id	integer			
lyrics_id	integer			
playlist_id	integer			
download_perm	boolean			

Indexes:

"songs_pkey" PRIMARY KEY, btree (song_id)

Foreign-key constraints:

"fk_song_artist_id" FOREIGN KEY (album_id) REFERENCES album(album_id)

"fk_song_genre_id" FOREIGN KEY (genre_id) REFERENCES genre(genre_id)

"fk_songs_lyrics_id" FOREIGN KEY (lyrics_id) REFERENCES lyrics(lyrics_id)

"fk_songs_playlist_id" FOREIGN KEY (playlist_id) REFERENCES playlist(playlist_id)

music_db=#