

```
CREATE DATABASE music WITH OWNER = postgres;
```

```
\c music;
```

```
CREATE TABLE IF NOT EXISTS list of performers (  
    performersId SERIAL PRIMARY KEY,  
    Name VARCHAR(80) NOT NULL UNIQUE  
);
```

```
CREATE TABLE IF NOT EXISTS Genre (  
    genreId SERIAL PRIMARY KEY,  
    Name VARCHAR(80) NOT NULL UNIQUE  
);
```

```
CREATE TABLE IF NOT EXISTS genrelist (  
    performersId INTEGER REFERENCES list of  
performers,  
    GenreId INTEGER REFERENCES Genre,  
    constraint pk_genrelist PRIMARY KEY  
(performersId, GenreId)  
);
```

```
CREATE TABLE IF NOT EXISTS Album (  
    albumId SERIAL PRIMARY KEY,  
    Name VARCHAR(80) NOT NULL UNIQUE,  
    Year INTEGER CHECK (Year > 1950 and Year < 2050)  
);
```

```
CREATE TABLE IF NOT EXISTS Albumsperformers (  
    performersId INTEGER REFERENCES list of  
performers,  
    AlbumId INTEGER REFERENCES Album,  
    constraint pk_Albumperformers PRIMARY KEY  
(performersId, AlbumId)  
);
```

```
CREATE TABLE IF NOT EXISTS Tracklist(  
    trackId SERIAL PRIMARY KEY,  
    AlbumId INTEGER REFERENCES Album,  
    Name VARCHAR(80) NOT NULL,
```

```
        Continuity INTEGER NOT NULL CHECK,  
        constraint pk_Tracklist UNIQUE (trackId, AlbumId)  
    );
```

```
CREATE TABLE IF NOT EXISTS Collection (  
    collectionId SERIAL PRIMARY KEY,  
    Name of the collection VARCHAR(80) NOT NULL,  
    Year INTEGER CHECK (Year > 1950 and Year < 2050)  
);
```

```
CREATE TABLE IF NOT EXISTS trackcollection (  
    CollectionId INTEGER REFERENCES Collections,  
    TrackId INTEGER REFERENCES Tracks,  
    constraint pk_trackcollection PRIMARY KEY  
(CollectionId, TrackId)
```