Conceptual Schema

Entities:

- Musician (SSN, Name, Address, Phone)
- Instrument (InstrumentID, Name, Key)
- Album (AlbumID, Title, CopyrightDate, Format, AlbumIdentifier)
- Song (SongID, Title, Author)

Relationships:

- Plays: Many-to-Many between Musician and Instrument
- Performs: Many-to-Many between Musician and Song
- Contained_In: One-to-Many from Album to Song
- Produces: One-to-One from Musician to Album (a musician can produce many albums, but an album has one producer)

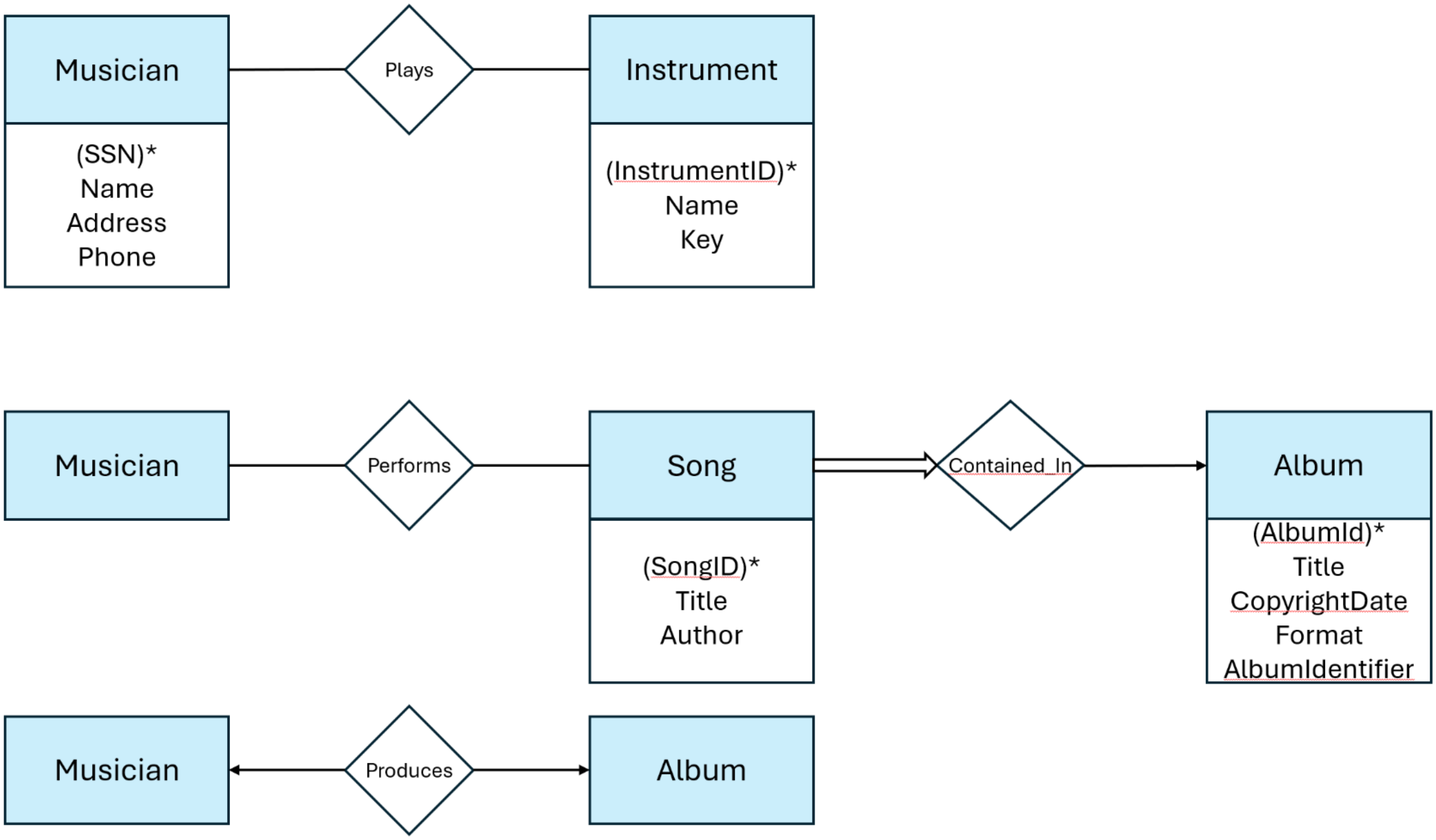
Assumptions:

- One address can belong to multiple musicians (so it's stored as a string, not a separate entity)
- A phone number may not be unique per musician (i.e., many phones per address), so treated separately or as a multi-valued attribute
- SongID added for uniqueness
- Each song appears on **only one** album
- Every album has **exactly one** producer

Uncaptured Constraints:

Some constraints cannot be fully captured in the ER diagram:

- I simplified multi-valued attributes like Phone or shared Address as non-unique strings.
- Though each Song is shown to belong to exactly one Album, additional logic in the database or application layer would be needed to enforce that it can't belong to multiple albums.


Musician
(SSN)*
Name
Address
Phone
Plays
Instrument
(InstrumentID)*
Name
Key
Musician
Performs
Song
(SongID)*
Title
Author
Contained_In
Album
(AlbumId)*
Title
CopyrightDate
Format
AlbumIdentifier
Musician
Produces
Album

Relational Schema

Table Name	Attributes
Musician	SSN (PK), Name, Address, Phone
Instrument	InstrumentID (PK), Name, Key
Album	AlbumID (PK), Title, CopyrightDate, Format, AlbumIdentifier, ProducerSSN (FK -> Musician.SSN)
Song	SongID (PK), Title, Author, AlbumID (FK -> Album.AlbumID)
Plays	SSN (FK -> Musician.SSN), InstrumentID (FK -> Instrument.InstrumentID), PRIMARY KEY (SSN, InstrumentID)
Performs	SSN (FK -> Musician.SSN), SongID (FK -> Song.SongID), PRIMARY KEY (SSN, SongID)