

CS 486/586 Introduction to DBMS

Fall 2021

Submitted by - Team 3: Agnita, Raghavendra

Assignment 4 – ER Diagram

Due: Friday, Oct 29, 11:59PM on Canvas

Please note the following:

- do this assignment in groups of 2.
- ensure that each group member's name is listed on the assignment to ensure credit
- submit your assignment in PDF format
- turn in your completed assignment on Canvas, one submission per group
- 100 points total

Part I - ER Diagrams and ER->Relations (100 pts)

You may use erdplus.com (free version) or [LucidChart](https://www.lucidchart.com) for completing these exercises.

Notes on erdplus.

- The erdplus GUI sometimes applies the cardinality symbol on the opposite entity than expected. Enter the data in erdplus so that the resulting diagram is correct.
- ***Regularly save your diagrams locally*** as a backup in case the website loses your account. It happens, I learned the hard way.

In this homework assignment, you're designing the database for an online fan-of-music database. You won't be building the entire online application, or accounting for all possible user interactions.

Background

There are many online sources for information about music, but much of the data isn't well structured, and you often have to go to multiple sites to get answers to your questions regarding musicians, bands, songs, albums, sales and concerts. When did Aerosmith form? Look in Wikipedia. When did Roger Waters leave Pink Floyd? Look on a different

page on Wikipedia. What songs are on Drake's album "*Take Care*"? Another page in Wikipedia, or Amazon, or iTunes. How many certified album sales does Beyonce have in the US? Yet another page in Wikipedia, or RIAA. When is Alicia Keys next giving a concert in Oregon or Washington? (September 1st at the Moda...)

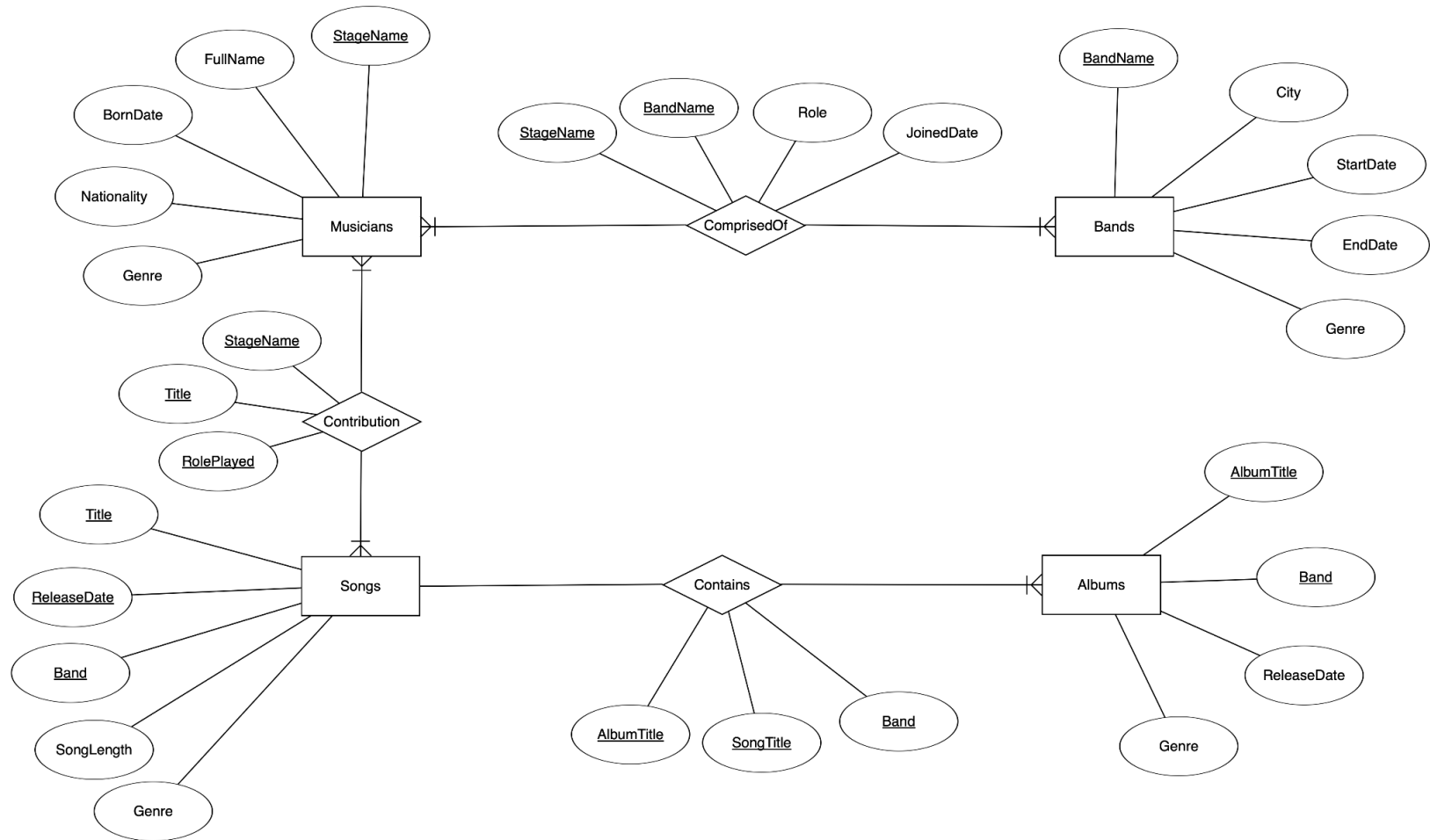
You will design a database schema that stores information about musicians, bands, songs, and albums. It won't contain all relevant attributes for each object, but enough to identify uniqueness constraints, foreign key constraints, and answer some of these interesting queries with one database.

Consider the following statements about your database:

- There are several core objects to be modeled in your database, including:
 - Musicians. Each musician has a unique stage name, full name, born date, and nationality
 - Bands. Each musical group is called a band, and is identified by their band name. It holds information about the city that they were formed in, the date the band was formed, and the date the band stopped being active (unless it's still active).
 - Songs. Each song is identified by a title, the date it was released, and the Band who released it. It also holds information about the length of the song.
 - Albums. Each album is identified by a title and the band who released it. It holds information about the date it was released.
- Note: Solo acts, like Lady Gaga, are modeled with both a Band named "Lady Gaga" and a musician named "Lady Gaga".
- Note: Album is the term used to release a set of music with a title, even if the music collection is released as a CD, or some other format.
- Here are the key relationships to model in your database...
 - Bands are comprised of Musicians. The database records the date when a musician joined a band, and what role they had in the band (such as Lead Vocals, Bass, Drums, etc)
 - Musicians contribute to Songs, whether or not they are in the Band that released the song. The db records what role the musician had in the song.
 - Albums contain Songs.
- The database has a list of Genre's, and tracks which Genre(s) are applicable to a Band, a Song, a Musician, or an Album.

1) Draw an ER diagram that represents these data requirements. Be sure to mark the key attributes and include cardinality constraints on relationships (1, many). Don't specify the types of attributes. (40 points)

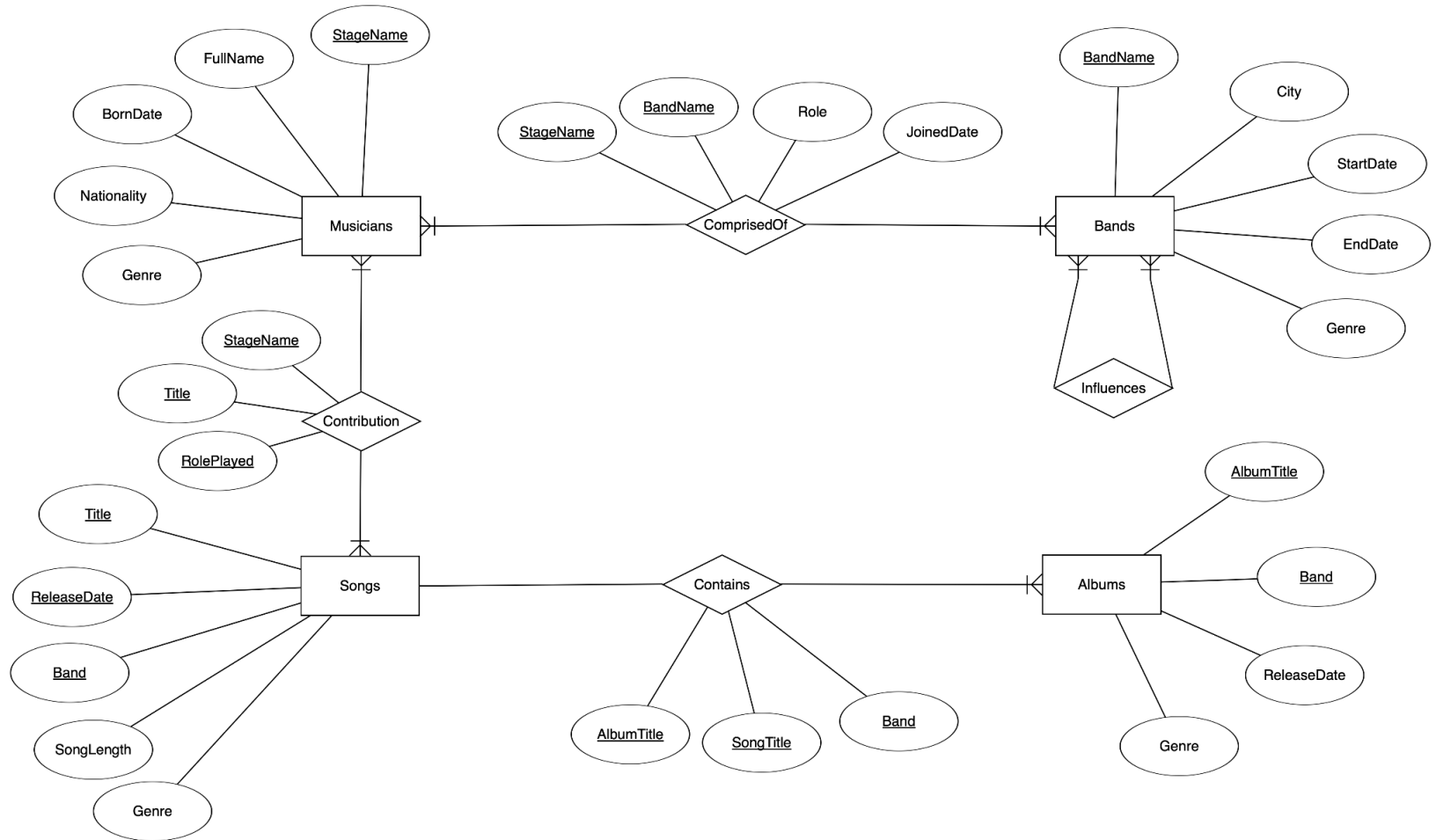
Solution:



2) Modify your ER diagram to handle the following extensions. Do each part as a separate change from the original diagram. You only need to show the parts of the diagram that change. (10 points)

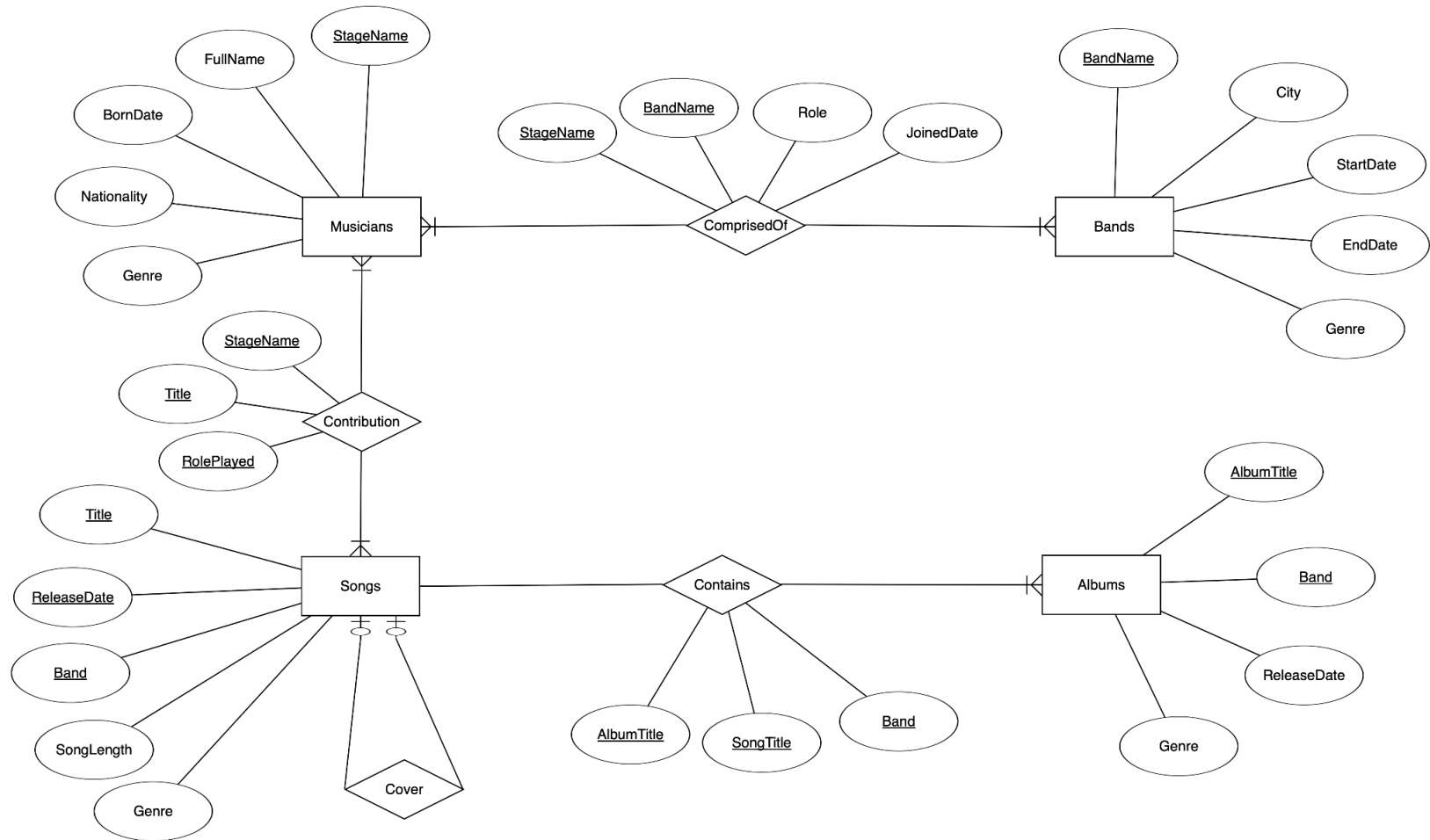
a) A band may have one or more influences, which are other bands.

Solution:



b) A song may be a cover of another (one) song.

Solution:



- 3) Translate Musicians, Bands, Songs and Albums (any any relations that tie those relations together) from your ERD into a db schema using the format:

*TableName1(Attribute1, Attribute2, Attribute3,...)
Attribute2 is a foreign key referencing Table3.*

Underline the attributes making up the primary key of each table. Describe foreign keys in a separate line. (10 points)

Solution:

Genre (GenreName)

Musicians (StageName, FullName, BornDate, Nationality, Genre)

Musicians.Genre is a foreign key that references Genre.GenreName

Bands (BandName, City, StartDate, EndDate, Genre)

Bands.Genre is a foreign key that references Genre.GenreName

ComprisedOf (StageName, BandName, Role, JoinedDate)

ComprisedOf.StageName is a foreign key that references Musicians.StageName

ComprisedOf.BandName is a foreign key that references Bands.BandName

Songs (Title, ReleaseDate, Band, SongLength, Genre)

Songs.Genre is a foreign key that references Genre.GenreName

Songs.Band is a foreign key that references Bands.BandName

Contribution (StageName, Title, RolePlayed)

Contribution.StageName is a foreign key that references Musicians.StageName

Contribution.Title is a foreign key that references Songs.Title

Albums (AlbumTitle, Band, ReleaseDate, Genre)

Albums.Genre is a foreign key that references Genre.GenreName

Albums.Band is a foreign key that references Bands.BandName

Contains (AlbumTitle, SongTitle, Band)

Contains.AlbumTitle is a foreign key that references Albums.AlbumTitle

Contains.SongTitle is a foreign key that references Songs.Title

Contains.Band is a foreign key that references Bands.BandName