Databases - Exercise 2: Modelling and Design

Sample solution

15 November 2019

1 Entity-Relationship Modelling

The domain to be modelled is music: artists, albums, songs, etc. The goal of this task could be to build a database for one's own music or a streaming service. We build the model in two steps. In both steps, your task is to build

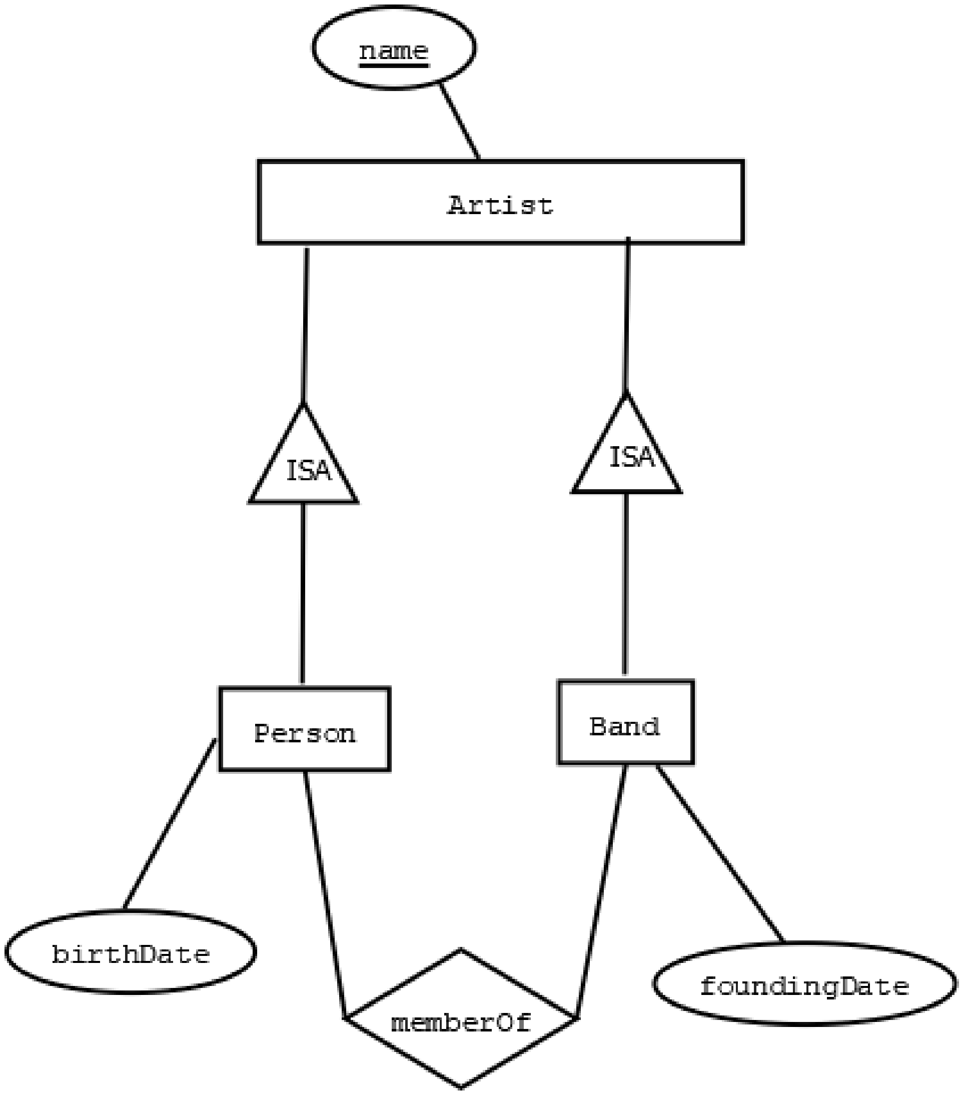
- an E-R diagram
- a database schema, either in SQL or in the usual schema notation

1.1 Step 1: artists and bands

Here are the main elements of the domain:

- Artists, uniquely identified by their names. An artist (as indicated by the author of a song or an album), can be
  - an individual person, who has a birth date
  - a band, which has a founding date
- An individual person can be a member in a band, also in several bands. Here we want the database to tell us who is a member of which bands.

Solution: ER after Step 1



Solution: Schema after Step 1

Artists(_name_)

Persons(_name_, birthDate)
name → Artists.name

Bands(_name_, foundingDate)
name → Artists.name

MemberOf(_personName_, _bandName_)
personName → Persons.name
bandName → Bands.name

1.2 Step 2: albums and songs

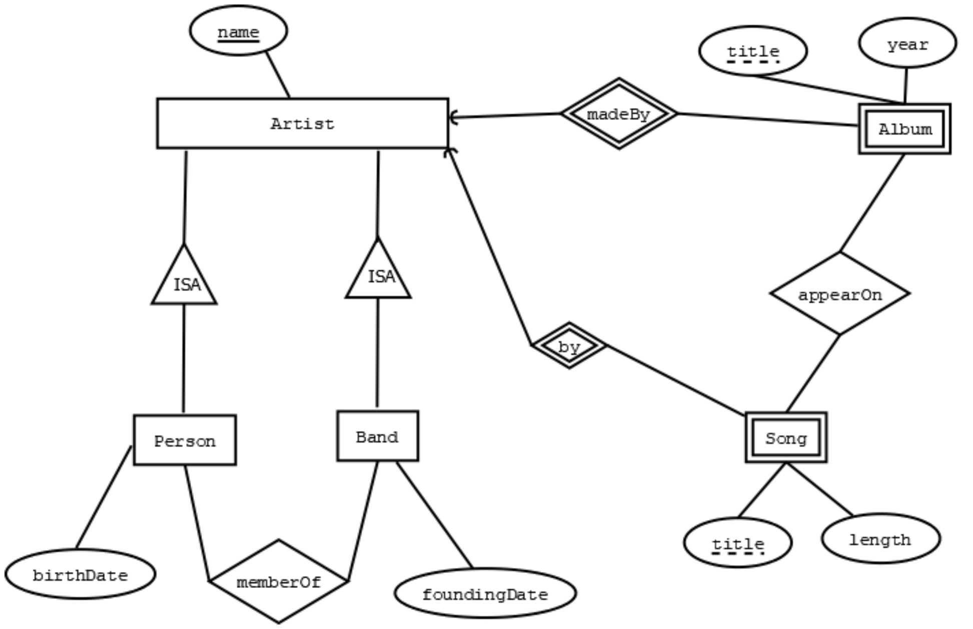
You should write the diagram for these elements beside or below the first diagram, but it should of course have arrows to the first diagram when appropriate.

- An album is identified by its title and the artist that made it, for instance, ”Greatest Hits” of Bob Dylan as opposed to ”Greatest Hits” by The Beatles. An album also has a year of appearance.

- A song is likewise identified by its title and the artist that made it. A song also has a length.

- A song can appear on several albums. The artist of the album can be different from the artist of the song, as for instance if the album is by ”Various Artists”. The database should be able to list, as separate rows, all songs that appear on different albums.

Solution: ER after Step 2



Solution: Schema after Step 2

Artists(_name_)

Persons(_name_, birthDate)
name → Artists.name

Bands(_name_, foundingDate)
name → Artists.name

MemberOf(_personName_, _bandName_)
personName → Persons.name
bandName → Bands.name

Albums(_title_, year, _artistName_)
artistName → Artists.name

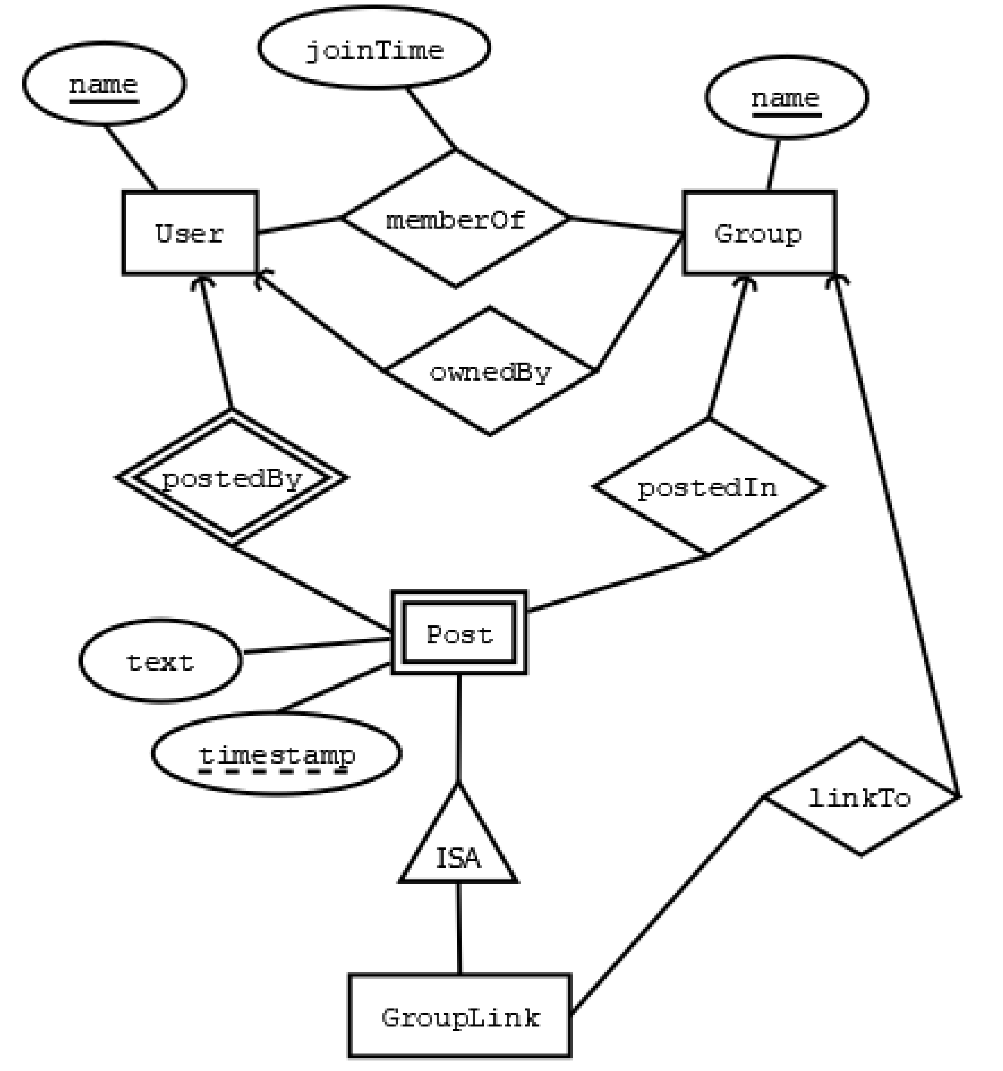
Songs(_title_, length, _artistName_)
artistName → Artists.name

appearOn(_songTitle_, _songArtistName_, _albumTitle_, _albumArtistName_)
(songTitle, songArtistName) → Songs.(title, artistName)
(albumTitle, albumArtistName) → Albums.(title, artistName)

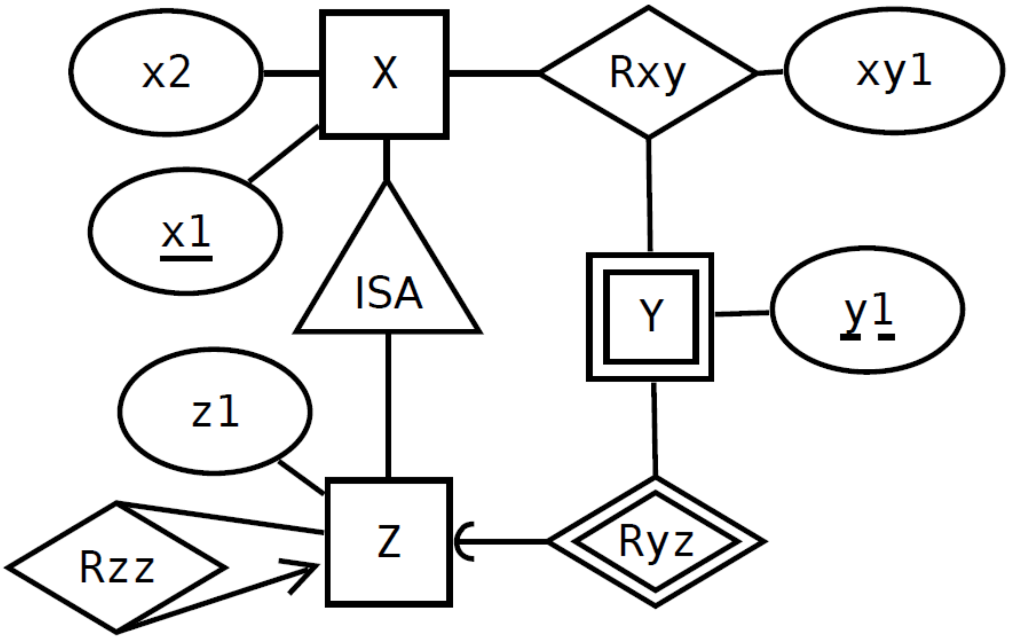
2 ER-design

a) Make an ER-diagram for a web application where users can register, join interest groups, and post messages to these groups. Users have unique user names, and groups have unique names. Users can be members of any number of groups. The time each user joined a group should be recorded. Each group has an owner, which is a user. Each post is posted in a group by a user. Posts contain text. Posts are identified by their timestamp and the username of the user posting it. There is also a special kind of post called a group link, these contain a link to a group in addition to the normal parts of a post.
You do not need to translate your diagram into a schema!

Solution: ER



b) Translate this (symbolic) ER-diagram into a relational schema, including keys and constraints.



Solution: Schema

X(_x1_, x2)

Z(_x1_, z1)
x1 → X.x1

Y(_y1_, _z_)
z → Z.x1

Rzz(_za_, _zb_)
za → Z.x1
zb → Z.x1

Rxy(_x_, _y_, _yz_, xy1)
x → X.x1
(y, yz) → Y.(y1, z)