Databases
TDA357/DIT621

Exercise 2

Entity-Relationship Modelling

1 Domain to model: Music

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/relational schema.

1.1 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.

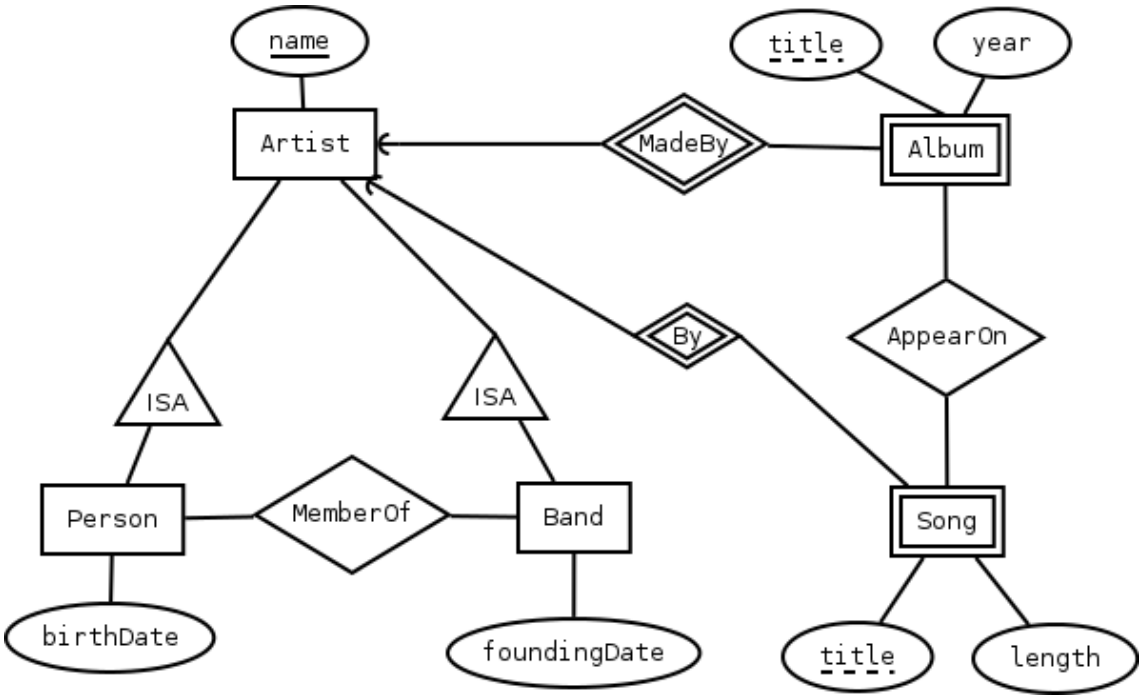
1.2 Albums and songs

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:



Artists (<u>name</u>)
Persons (<u>name</u>, birthDate)
  name → Artists.name
Bands (<u>name</u>, foundingDate)
  name → Artists.name
MembersOf (<u>personName</u>, <u>bandName</u>)
  personName → Persons.name
  bandName → Bands.name

Albums (<u>title</u>, year, <u>artistName</u>)
  artistName → Artists.name
Songs (<u>title</u>, length, <u>artistName</u>)
  artistName → Artists.name
AppearsOn (<u>songTitle</u>, <u>songArtistName</u>, <u>albumTitle</u>, <u>albumArtistName</u>)
  (songTitle, songArtistName) → Songs.(title, artistName)
  (albumTitle, albumArtistName) → Albums.(title, artistName)

2 Domain to model: Users, groups and posts

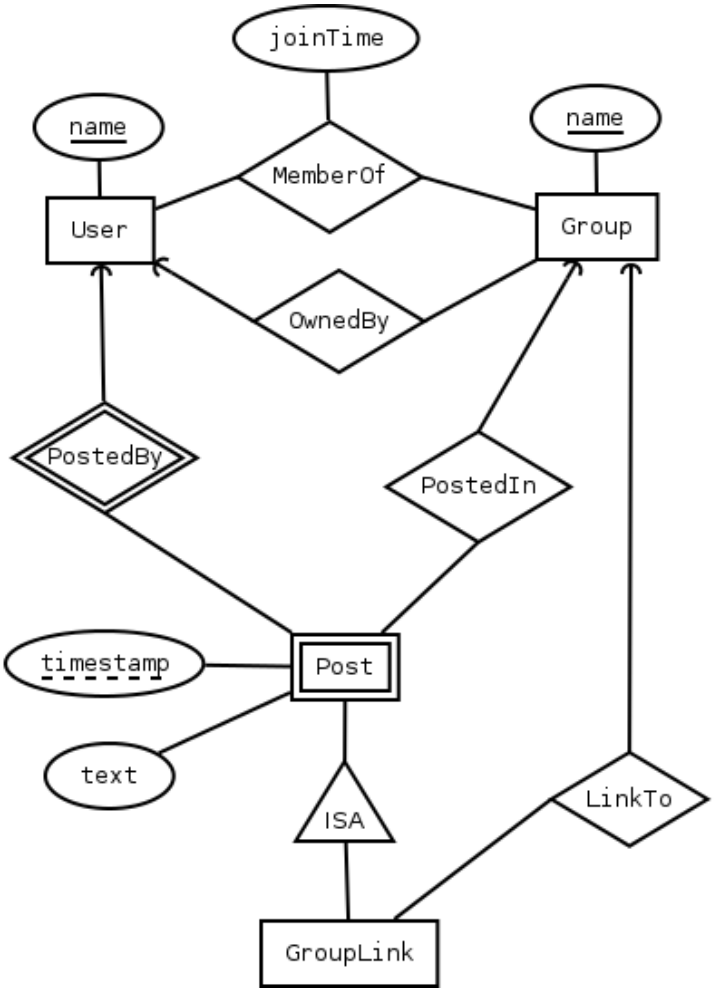
a) Make an ER-diagram for a web application where users can register, join interest groups, and post messages to these groups.

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.

b) Translate your diagram into a schema.

Solution:



Users (name)
Groups (name, owner)
  owner → Users.name
MembersOf (uname, gname, joinTime)
  uname → Users.name
  gname → Groups.name
Posts (uname, timestamp, text, gname)
  uname → Users.name
  gname → Groups.name
GroupLinks (uname, timestamp, gname)
  (uname, timestamp) → Posts. (uname, timestamp)
  gname → Groups.name

3 Domain to model: Employees at a university

(Adapted from a previous exam question.)

a) You will make an ER-diagram for the employee structure of a university.

   The university is divided into departments. Departments in turn are divided into multiple divisions, which in turn are divided into multiple units.

   Departments, divisions, and units are identified by name with (only) the following limitations:

   - No two departments can have the same name.
   - There cannot be two divisions with the same name within a department.
   - There cannot be two units with the same name within a division.

   An example of a unit would be the "Functional Programming" unit of the "Computing Science" division of the "Department of Computer Science and Engineering".

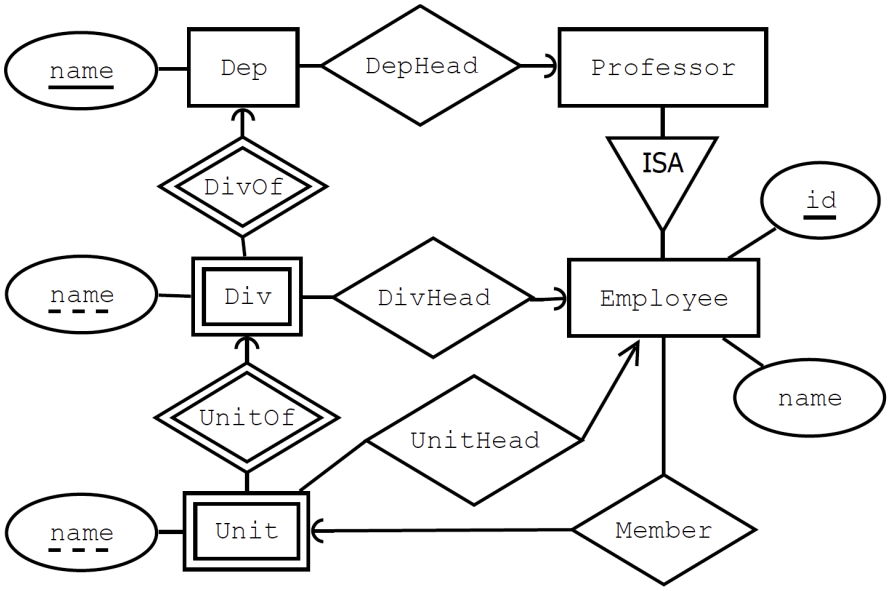
   Each employee is a member of a unit (and thus indirectly of a division and department). Each employee has a name and an identifying number.

   Employees can be designated as heads of departments, heads of divisions and heads of units, with the following rules:

   - Each department and division must have a single head.
   - Each head of department must be a professor (and only some of the employees are professors).
   - Units may or may not have a head (units that have no head are handled by the head of division).

b) Translate your diagram into a schema.

Solution:

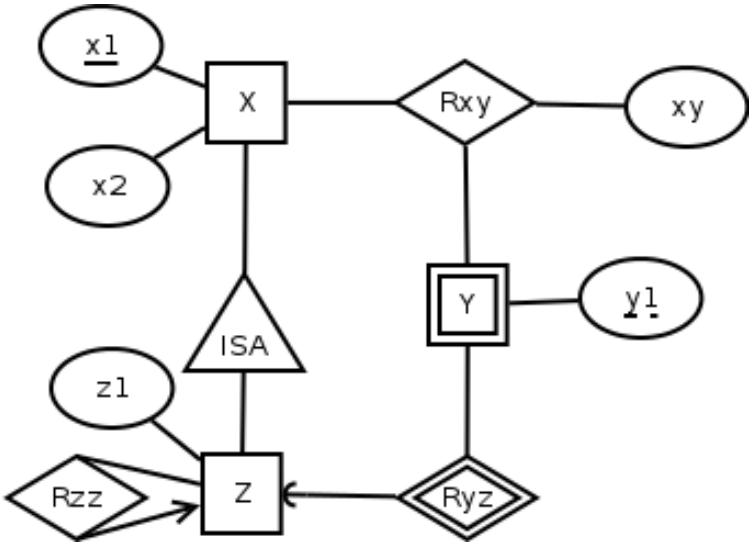


Employees (id, name, depname, divname, unit)
  (depname, divname, unit) → Units(depname, divname, unit)
Professors (id)
  id → Employees.id
Deps (depname, head)
  head → Professors.id
Divs (depname, divname, head)
  depname → Deps.depname
  head → Employees.id
Units (depname, divname, unit)
  (depname, divname) → Divs.(depname, divname)
UnitHeads (depname, divname, unit, id)
  id → Employees.id
  (depname, divname, unit) → Units.(depname, divname, unit)

4 Diagram to schema

Translate each of the (symbolic) ER-diagrams into a relational schema, including keys and constraints.

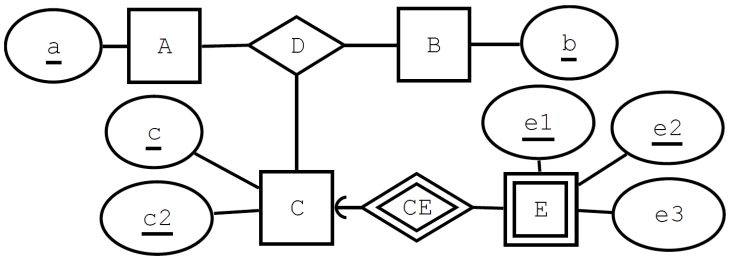
a)



Solution:

Xs (<u>x1</u>, x2)
Zs (<u>x1</u>, z1)
  x1 → Xs.x1
Ys (<u>y1</u>, <u>z</u>)
  z → Zs.x1
Rzzs (<u>za</u>, zb)
  za → Zs.x1
  zb → Zs.x1
Rxys (<u>x</u>, <u>y</u>, <u>yz</u>, xy)
  x → Xs.x1
  (y, yz) → Ys.(y1, z)

b) (From a previous exam question.)



Solution:

As (a)
Bs (b)
Cs (c,c2)
Ds (a, b, c, c2)
a → As.a
b → Bs.b
(c,c2) → Cs.(c,c2) Es (c, c2, e1, e2, e3)
(c,c2) → Cs.(c, c2)