Introduction to Database Systems

Database for music related data

Szűcs Benedek
(BGP18S)

Project Supervisor:
Gábor Csaba

Szűcs Benedek
BGP18S

Introduction

I would like to create a small database to show examples of all types of entities and relationships. I will store data about my favourite musicians and their work, including information about released albums, singles, record labels, bands.

Description

People
In general their names, birth dates and home countries will be stored however, there will be several subclasses with specific connections: musicians will have their bands stored and the post on which they play. Managers of record labels will be connected to the companies they own.

Bands
The name of the band, the year in which the band formed, their country, music genre will be stored. Also their releases and members will be connected to them.

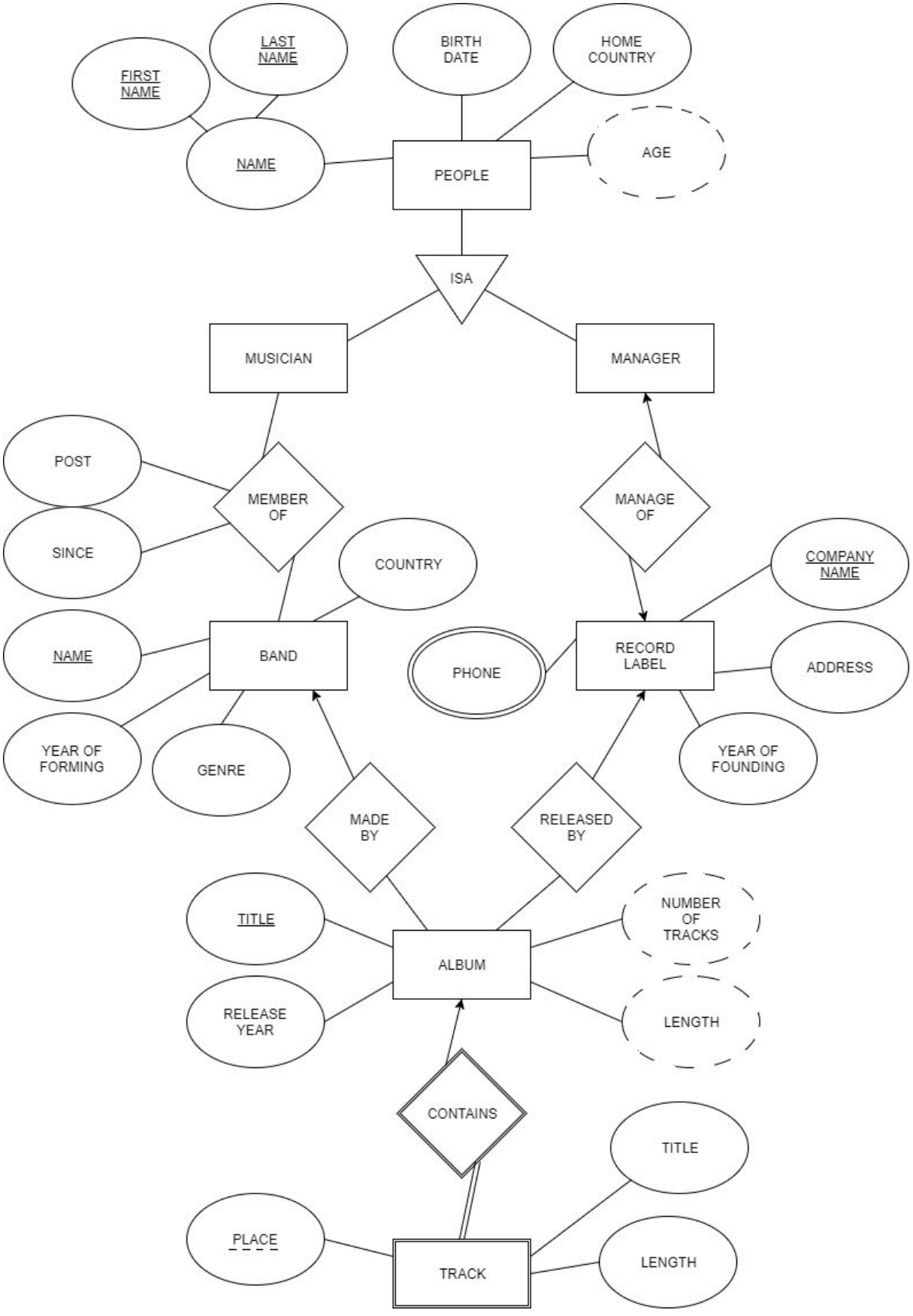
Albums
Albums will be connected to the band and the releasing record label. Title, production year, length, number of tracks will be stored for each release.

Tracks
A track will be described by its title, length and its place on the album. Tracks will be connected to the album on which they can be found.

Record labels
I will store the names, countries of the headquarters and founding dates of record labels. They will be connected to the albums they released and the manager of the company.

Szűcs Benedek
BGP18S

E/R Diagram



Szűcs Benedek
BGP18S

Relationships and ISA-s

- musician ISA subclass of people
- manager ISA subclass of people
- musician and band have a *many-tomany* relationship
- manager and record label have a *one-to-one* relationship
- band and album have a *one-to-many* relationship
- record label and album have a *one-to-many* relationship
- track is a weak entity and has a *weak* relationship with album

Relational model

PEOPLE (first_name, last_name, dirth_date, country, profession)
BAND (band_name, genre, country, forming_year)
MEMBER_OF (band_name, first_name, last_name, post, since_year)
RECORD_LABEL (label_name, address, founding_year, first_name, last_name)
LABEL_CONTACT (label_name, phone_number)
ALBUM(album_title, release_year, band_name, label_name)
TRACK (album_title, place, track_title, length)

Normal forms

All tables are in BCNF, because all attributes are dependent on the primary key, but not on any other attribute.

Creating tables

```
create table PEOPLE(
  FIRST_NAME varchar2(40),
  LAST_NAME varchar2(20),
  BIRTH_DATE date,
  BIRTH_COUNTRY varchar2(50),
  PROFESSION varchar2(8) not null check(PROFESSION='musician' or
PROFESSION='manager'),
  primary key(FIRST_NAME, LAST_NAME));

create table BAND(
  BAND_NAME varchar2(50),
  GENRE varchar2(30) not null,
  COUNTRY varchar2(50),
  FORMING_YEAR number(4),
```

```
  primary key(BAND_NAME));

create table MEMBER_OF(
  BAND_NAME varchar2(50),
  FIRST_NAME varchar2(40),
  LAST_NAME varchar2(20),
  POST varchar2(20) not null,
  SINCE number(4),
  primary key(BAND_NAME, FIRST_NAME, LAST_NAME),
  foreign key(BAND_NAME) references BAND(BAND_NAME),
  foreign key(FIRST_NAME, LAST_NAME) references PEOPLE(FIRST_NAME,
LAST_NAME));

create table RECORD_LABEL(
  LABEL_NAME varchar2(50),
  ADDRESS varchar2(300) not null,
  FOUNDING_YEAR number(4),
  FIRST_NAME varchar2(40),
  LAST_NAME varchar2(20),
  primary key(LABEL_NAME),
  foreign key(FIRST_NAME, LAST_NAME) references PEOPLE(FIRST_NAME,
LAST_NAME));

create table LABEL_CONTACT(
  LABEL_NAME varchar2(50),
  PHONE_NUMBER number(20),
  primary key(PHONE_NUMBER),
  foreign key(LABEL_NAME) references RECORD_LABEL(LABEL_NAME));
alter table LABEL_CONTACT drop primary key;
alter table LABEL_CONTACT drop column PHONE_NUMBER;
alter table LABEL_CONTACT add CONTACT varchar2(50);
alter table LABEL_CONTACT add primary key(CONTACT);

create table ALBUM(
  ALBUM_TITLE varchar2(50),
  RELEASE_YEAR number(4) not null,
  BAND_NAME varchar2(50),
  LABEL_NAME varchar2(50) default 'self release',
  primary key(ALBUM_TITLE),
  foreign key(BAND_NAME) references BAND(BAND_NAME),
  foreign key(LABEL_NAME) references RECORD_LABEL(LABEL_NAME));
```

```
create table TRACK(
  ALBUM_TITLE varchar2(50),
  PLACE number(2),
  TRACK_TITLE varchar2(50) not null,
  LENGTH number(4) not null,
  primary key(ALBUM_TITLE, PLACE),
  foreign key(ALBUM_TITLE) references ALBUM(ALBUM_TITLE));
```

Simple queries

List the full names of people who were born before 1980.
Adja meg azon emberek teljes nevét, akik 1980 előtt születtek.

$$\pi_{\text{FIRST\_NAME, LAST\_NAME}}(\sigma_{\text{BIRTH\_DATE < 1980-01-01}}(\text{PEOPLE}))$$

```
select FIRST_NAME, LAST_NAME
from PEOPLE
where BIRTH_DATE < date '1980-01-01';
```

```
Corey    Taylor
Joey    Jordison
```

List the full names and posts of musicians who play multiple instruments.
Adja meg azon zenészek teljes nevét és hangszereit, akik több hangszeren is játszanak.

$$\pi_{\text{M1.FIRST\_NAME, M1.LAST\_NAME, M1.POST}}(\sigma_{\text{M1.FIRST\_NAME = M2.FIRST\_NAME} \wedge \text{M1.LAST\_NAME = M2.LAST\_NAME} \wedge \neg\text{M1.POST = M2.POST}}(\rho_{\text{M1}}(\text{MEMBER\_OF}) \times \rho_{\text{M2}}(\text{MEMBER\_OF})))$$

```
select M1.FIRST_NAME, M1.LAST_NAME, M1.POST
from MEMBER_OF M1, MEMBER_OF M2
where M1.FIRST_NAME = M2.FIRST_NAME and
     M1.LAST_NAME = M2.LAST_NAME and
     M1.POST != M2.POST;
```

```
Joey    Jordison    lead guitar
Joey    Jordison    drums
```

List the full names and bands of musicians who play in multiple groups.

Adja meg azon zenészek teljes nevét és zenekarait, akik több bandában is játszanak.

$$\pi_{M1.FIRST\_NAME,\ M1.LAST\_NAME,\ M1.BAND\_NAME}(\sigma_{M1.FIRST\_NAME = M2.FIRST\_NAME \wedge M1.LAST\_NAME = M2.LAST\_NAME \wedge \neg M1.BAND\_NAME = M2.BAND\_NAME}(\rho_{M1}(MEMBER\_OF) \times \rho_{M2}(MEMBER\_OF)))$$

```
select M1.FIRST_NAME, M1.LAST_NAME, M1.BAND_NAME
from MEMBER_OF M1, MEMBER_OF M2
where M1.FIRST_NAME = M2.FIRST_NAME and
    M1.LAST_NAME = M2.LAST_NAME and
    M1.BAND_NAME != M2.BAND_NAME
order by M1.LAST_NAME;
```

Joey Jordison Slipknot
Joey Jordison Murderdolls
Corey Taylor Stone Sour
Corey Taylor Slipknot

List the full names and bands of musicians who were not founders of the group.
Adja meg azon zenészek teljes nevét és zenekarát, akik nem voltak alapító tagok az adott bandában.

$$\pi_{MEMBER\_OF.FIRST\_NAME,\ MEMBER\_OF.LAST\_NAME,\ MEMBER\_OF.BAND\_NAME}(\sigma_{MEMBER\_OF.SINCE > BAND.FORMING\_YEAR \wedge MEMBER\_OF.BAND\_NAME = BAND.BAND\_NAME}(MEMBER\_OF \times BAND))$$

```
select MEMBER_OF.FIRST_NAME, MEMBER_OF.LAST_NAME,
MEMBER_OF.BAND_NAME
from MEMBER_OF, BAND
where MEMBER_OF.SINCE > BAND.FORMING_YEAR and
    MEMBER_OF.BAND_NAME = BAND.BAND_NAME;
```

Corey Taylor Slipknot

List all the albums and their lengths.
Adja meg az albumokat a hosszukkal együtt.

$$_{ALBUM\_TITLE,}\mathbf{g}_{sum(LENGTH)}(TRACK)$$

```
select ALBUM_TITLE, floor(sum(LENGTH)/60) MINUTE,
sum(LENGTH)-floor(sum(LENGTH)/60)*60 SECOND
from TRACK
group by ALBUM_TITLE;
```

With Roots Above and Branches Below 40 21
Deep Blue 43 38

Get the album and the releasing label with the longest average tracks.
Adja meg az albumot, ami átlagosan a leghosszabb számokat tartalmazza és a hozzá tartozó kiadót.

$$\Pi_{\text{TRACK.ALBUM\_TITLE, ALBUM.LABEL\_NAME, TRACK.ALBUM\_TITLE, ALBUM.LABEL\_NAME}}\mathbf{g}_{\text{avg(LENGTH)}}$$
$$(\sigma_{\text{TRACK.ALBUM\_TITLE = ALBUM.ALBUM\_TITLE}}(\text{TRACK}\mathbf{X}\text{ALBUM}))$$

```
select TRACK.ALBUM_TITLE, ALBUM.LABEL_NAME, avg(LENGTH) as AV_LN
from TRACK, ALBUM
where TRACK.ALBUM_TITLE = ALBUM.ALBUM_TITLE
group by TRACK.ALBUM_TITLE, ALBUM.LABEL_NAME
order by AV_LN desc;
```

With Roots Above and Branches Below Ferret Music
220,72727272727272727272727272727273

List labels with multiple contacts.
Adja meg azokat a kiadókat, akiknek több elérhetőség is meg van adva.

$$\Pi_{\text{LABEL\_NAME}}(\sigma_{\text{C > 1}}(\Pi_{\text{LABEL\_NAME, LABEL\_NAME}}\mathbf{g}_{\text{count(CONTACT)->C}}(\text{LABEL\_CONTACT})))$$

```
select LABEL_NAME
from (select LABEL_NAME, count(CONTACT) C
    from LABEL_CONTACT
    group by LABEL_NAME)
where C > 1;
```

Ferret Music

List all bands and their genres that fall into some kind of metal genre.

Adja meg a valamilyen metal stílusba eső zenekarok neveit és stílusát.

$\pi_{BAND\_NAME,\ GENRE}(\sigma_{GENRE\ like\ '\%metal\%'}(BAND))$

```
select BAND_NAME, GENRE
from BAND
where GENRE like '%metal%';
```

The Devil Wears Prada metalcore
Slipknot heavy metal
Stone Sour heavy metal
Murderdolls glam metal
Parkway Drive metalcore

Select bands with a lead guitarist and a keyboard player.
Válassza ki a zenekarokat, amikben szólógitáros és billentyűs is játszik.

$\pi_{BAND\_NAME,\ POST}(MEMBER\_OF) \div \pi_{POST}(\sigma_{POST=lead\ guitar\ \vee\ POST=keyboard}(MEMBER\_OF))$

```
select BAND_NAME
from BAND
where BAND_NAME in ((select BAND_NAME
                     from MEMBER_OF
                     where POST like 'keyboard')
                     intersect
                     (select BAND_NAME
                     from MEMBER_OF
                     where POST like 'keyboard'));
```

The Devil Wears Prada

List the tracks from the album Deep Blue which are longer than any track on the album With Roots Above and Branches Below.
Válassza ki azokat a számokat a Deep Blue albumról, amik hosszabbak bármely számnál a With Roots Above and Branches Below albumon.

$\pi_{TRACK\_TITLE}(\sigma_{ALBUM\_TITLE=Deep\ Blue\ \wedge\ LENGTH>\mathcal{g}_{max(LENGTH)}(\sigma_{ALBUM\_TITLE=With\ Roots\ Above\ and\ Branches\ Below}(TRACK))(TRACK))$

```
select TRACK_TITLE
```

```
from TRACK
where ALBUM_TITLE like 'Deep Blue' and
    LENGTH > all(select LENGTH
            from TRACK
            where ALBUM_TITLE like 'With Roots Above and Branches Below');
```

Alone

Select bands which were formed before any of the record labels were founded.
Válassza ki azokat a zenekarokat, amelyikek régebbiek valamelyik kiadónál.

π_{BAND_NAME}
$(\sigma_{FORMING\_YEAR<\mathbf{g}_{max(FOUNDING\_YEAR)}}(\pi_{FOUNDING\_YEAR}(RECORD\_LABEL))(BAND))$

```
select BAND_NAME
from BAND
where FORMING_YEAR < any(select FOUNDING_YEAR
                from RECORD_LABEL);
```

Stone Sour
Slipknot

We want to store all subgenres of rock music labeled as “rock”. Update the database accordingly.

```
update BAND
set GENRE = 'rock'
where GENRE like '%metal%' or
    GENRE like '%rock%';
```

Joey Jordison retired. Remove him from all bands.

```
delete from MEMBER_OF
where FIRST_NAME like 'Joey' and
    LAST_NAME like 'Jordison';
```

Create a view with all music produced by Ferret Music.

```
create view FERRET_TRACKS as
select TRACK_TITLE, LENGTH
from TRACK, ALBUM
```

```
where TRACK.ALBUM_TITLE = ALBUM.ALBUM_TITLE and
      ALBUM.LABEL_NAME like 'Ferret Music';
```