

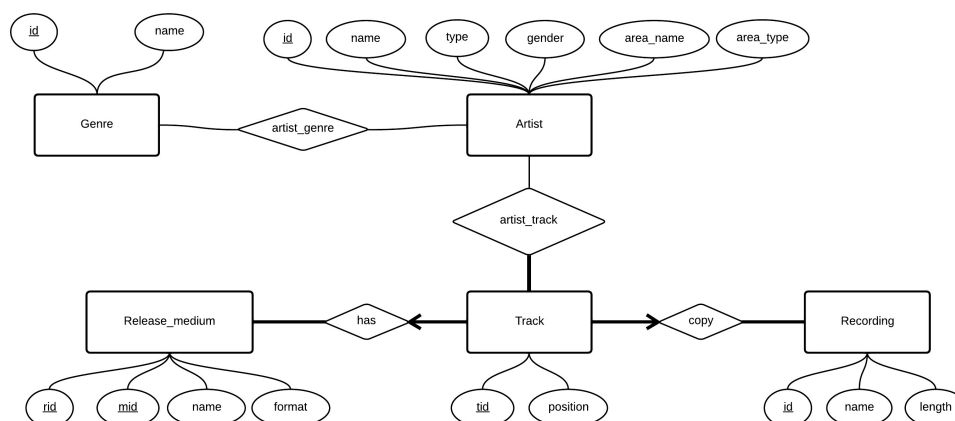
Project Report

Group 18: Chengzhen Wu, Huiqi Mao, Ruofan Zhou

May 8, 2014

1 ER model

After reading the feedback of diverable1, we modified our ER model as below:



In the given project data, we firstly recognize 'area', 'artist' and 'genre' each as three individual entities. Each artist is from at most one area, so it's a many-to-one relation. Several artists can belong to different genres and one genre can contain several artists. So the relation between 'artist' and 'genre' is many-to-many.

Secondly, we think about the relationship among 'release', 'recording', 'track' and 'medium'. We imagine a scene to describe these relations. The csv file of 'release' contains the names of releases. They could be stored in the mediums, such as CD, 12" Vinyl and so on. What's more, one release could have several CDs to contain many tracks, or in different medium (I'm not sure about this, but possible). So the relation between 'release' and 'medium' is one-to-many. Next, each track in different mediums must correspond to one recording. So the relation between 'track' and 'recording' is many-to-one. Each 'track' must be in one of 'medium's. So the relation is many-to-one.

Finally, we merge one-to-many relations. 'release' and 'medium' are merged into 'Release_medium'. 'area' can be merged into 'artist' as attributes. The 'has' relation between 'Release_medium' and 'track' is merged into 'track' using 'mid' as foreign key. So is the 'copy' relation between 'recording' and 'track' using 'id' of recording as foreign key.

Additionally, we ignore the 'count' in 'genre', which could be created as view in the database.

2 SQL based on ER model

```
--ENTITY Genre
--we don't need gcount in genre
CREATE TABLE Genre (
  GID INTEGER,
  Gname VARCHAR(100),
  PRIMARY KEY (GID)
);

--ENTITY Artist
CREATE TABLE Artist (
  AID INTEGER,
  Aname VARCHAR(100),
  Atype CHAR(6),
  gender CHAR(6),
  area_name VARCHAR(60),
  area_type CHAR(15)
  PRIMARY KEY (AID)
);

--ENTITY Recording
CREATE TABLE recording (
  RID INTEGER,
  Rname VARCHAR(100),
  Rlength INTEGER,
  PRIMARY KEY (RID)
);

--ENTITY ReleaseMedium
CREATE TABLE ReleaseMedium (
  MID INTEGER,
  RID INTEGER,
  name VARCHAR(400),
  format CHAR(45),
  PRIMARY KEY (RID,MID)
);

--ENTITY Track
CREATE TABLE Track (
  TID INTEGER,
  position INTEGER,
  MID INTEGER NOT NULL,
  RID INTEGER NOT NULL,
  REID INTEGER NOT NULL,
  PRIMARY KEY (TID),
  FOREIGN KEY (RID,MID) REFERENCES ReleaseMedium ON DELETE NO ACTION,
  FOREIGN KEY (REID) REFERENCES Recording ON DELETE NO ACTION
);

--RELATIONSHIP artist_genre
CREATE TABLE artist_genre (
  AID INTEGER NOT NULL,
  GID INTEGER NOT NULL,
  PRIMARY KEY (AID, GID),
  FOREIGN KEY (AID) REFERENCES Artist ON DELETE NO ACTION ,
  FOREIGN KEY (GID) REFERENCES Genre ON DELETE NO ACTION
);

--RELATIONSHIP artist-track
CREATE TABLE artist_track (
  AID INTEGER NOT NULL,
  TID INTEGER NOT NULL,
```

```
PRIMARY KEY (AID, TID),
FOREIGN KEY (AID) REFERENCES Artist ON DELETE NO ACTION ,
FOREIGN KEY (TID) REFERENCES Track ON DELETE NO ACTION
);
```

3 Alternative based on Real Data

When importing the real data, we find several problems. So we decide to change the schema in order to import as much data as possible.

In our ER model, we merged the 'area' and 'artist'. However, since the data is not incomplete, a few AreaIDs in 'artist' don't appear in 'area', such as 'AreaID=241'. So we use an 'area' entity instead. And the foreign key in 'artist' is disabled. Other foreign key constraints, except in 'medium', are also disabled to make it easy to import data. Additionally, in the original file of 'artist.csv', we change 'N' to '0', which is null in the 'area' table.

Another problem with the data is duplicate, since the data in artist_track has duplicate, we set the foreign-key disabled in the table artist_track.

We separate the release_medium table into 'release' and 'medium', considering the limited space. Because the name of release could be very long. Duplication could cause a waste of memory. But the foreign key in 'medium' is still retained, because it works.

In the 'genre' entity, 'count' is remained for further possible queries(We find that 'count' from real data is not really the count we calculate).

4 Alternative SQL

```
CREATE TABLE Init_Genre (
  GID VARCHAR(20),
  Gname VARCHAR(300),
  count Integer,
  PRIMARY KEY (GID)
);

create table Init_Area(
  AreaID VARCHAR(20),
  AreaName VARCHAR(150),
  Area_type VARCHAR(15),
  PRIMARY KEY (AreaID)
);

CREATE TABLE Init_Artist (
  AID VARCHAR(20),
  Aname VARCHAR(200),
  Atype CHAR(10),
  gender CHAR(6),
  --area_name VARCHAR(100),
  --area_type CHAR(15),
  AreaID VARCHAR(20),
  PRIMARY KEY (AID),
  foreign key (AreaID) references Init_area(AreaID)
);
```

```

CREATE TABLE Init_Recording (
  RID VARCHAR(20),
  Rname VARCHAR(2000),
  Rlength VARCHAR(20),
  PRIMARY KEY (RID)
);

CREATE TABLE Init_Release (
  RID VARCHAR(20),
  name VARCHAR(1000),
  PRIMARY KEY (RID)
);

CREATE TABLE Init_Medium (
  MID VARCHAR(20),
  RID VARCHAR(20),
  format CHAR(45),
  PRIMARY KEY (MID),
  FOREIGN KEY (RID) REFERENCES Init_Release (RID)
);

CREATE TABLE Init_Track (
  TID VARCHAR(20),
  REID VARCHAR(20) NOT NULL,
  MID VARCHAR(20) NOT NULL,
  position INTEGER,
  PRIMARY KEY (TID),
  FOREIGN KEY (MID) REFERENCES Init_Medium (MID),
  FOREIGN KEY (REID) REFERENCES Init_Recording (RID)
);

CREATE TABLE Init_artist_genre (
  AID VARCHAR(20) NOT NULL,
  GID VARCHAR(20) NOT NULL,
  PRIMARY KEY (AID, GID),
  FOREIGN KEY (AID) REFERENCES Init_Artist (AID) ON DELETE CASCADE,
  FOREIGN KEY (GID) REFERENCES Init_Genre (GID) ON DELETE CASCADE
);

CREATE TABLE Init_artist_track (
  AID VARCHAR(20) NOT NULL,
  TID VARCHAR(20) NOT NULL,
  PRIMARY KEY (AID, TID),
  FOREIGN KEY (AID) REFERENCES Init_Artist (AID), --these foreign keys are disabled for data duplicate
  FOREIGN KEY (TID) REFERENCES Init_Track (TID)
);

```

5 Queries

We finished the queries based on our model:

A

```

select artist.ANAME from init_area area, init_artist artist
      where area.areaname='Switzerland';

```

B

```

create view area_male as
select area.AREAID, area.areaname, count(*) as sum
from INIT_Area area, INIT_ARTIST artist
where artist.AREAID = area.AREAID and area.AREAID <> '0' and artist.GENDER='Male'
GROUP BY area.areaID, area.Areaname
order by count(*) desc;

```

```

create view area_female as
select area.AREAID,area.areaname,count(*) as sum
from INIT_Area area, INIT_ARTIST artist
where artist.AREAID = area.AREAID and area.AREAID<>'0'
and artist.GENDER='Female'
GROUP BY area.areaID, area.areaname
order by count(*) desc;

create view area_group as
select area.AREAID,area.areaname,count(*) as sum
from INIT_Area area, INIT_ARTIST artist
where artist.AREAID = area.AREAID
and area.AREAID<>'0'
and artist.atype='Group'
GROUP BY area.areaID, area.areaname
order by count(*) desc;

select area_male.areaname,area_male.sum as MALE,
area_female.sum as FEMALE,area_group.sum as GROU
from AREA_MALE,AREA_GROUP,AREA_FEMALE
where (AREA_FEMALE.AREAName=AREA_MALE.AREAName and AREA_FEMALE.AREAName=AREA_
    from area_female
    where ROWNUM=1))or
    (AREA_MALE.AREAName=AREA_FEMALE.AREAName
    and AREA_MALE.AREAName=AREA_Group.AREAName
    and AREA_MALE.AREANAME=(select area_male.areaname
    from area_male
    where ROWNUM=1))or
    (AREA_GROUP.AREAName=AREA_MALE.AREAName
    and AREA_FEMALE.AREAName=AREA_Group.AREAName
    and AREA_GROUP.AREANAME=(select area_group.areaname
    from area_group
    where ROWNUM=1));

```

C

```

select artist.aname
from (
select artist.ANAME,count(*)
    from init_artist artist,init_artist_track A_T
    where artist.atype='Group' and A_T.aid=artist.aid
    group by artist.aname order by count(*) desc) artist
where Rownum<11;

```

D

```
select art.ANAME
from (select art.aid
      from INIT_ARTIST art
      join INIT_ARTIST_TRACK art_track
        on art.AID = art_track.AID
      join INIT_TRACK track on art_track.TID = track.TID
      join INIT_MEDIUM medium on track.MID = medium.MID
      join INIT_RELEASE release on medium.RID = release.RID
     where art.atype = 'Group'
     group by art.aid
     order by count(*) desc) info, init_ARTIST art
WHERE art.AID = info.AID and rownum <= 10
ORDER BY rownum;
```

E

```
select art1.aname
      from (select art.aid
            from init_artist art, init_artist_genre
            art_genre, init_genre genre
            where art.aid = art_genre.aid
            and art_genre.GID = genre.GID
            and art.gender = 'Female'
            group by art.aid
            order by count(*))
genreRank, init_artist art1
where genreRank.aid = art1.aid and rownum = 1;
```

F

```
select area.areaname
      from INIT_AREA area
     where area.area_type='City' and (select count(*)
      from INIT_ARTIST artist
     where artist.AREAID = area.areaID
    and artist.GENDER='Male')<(select count(*)
    from INIT_ARTIST artist,INIT_AREA area
   where artist.AREAID = area.areaID and artist.GENDER='Female')
```

G

```
create view med_track as
      select medium.MID,count(*) as tracks
      from INIT_Medium medium,INIT_TRACK track
```

```

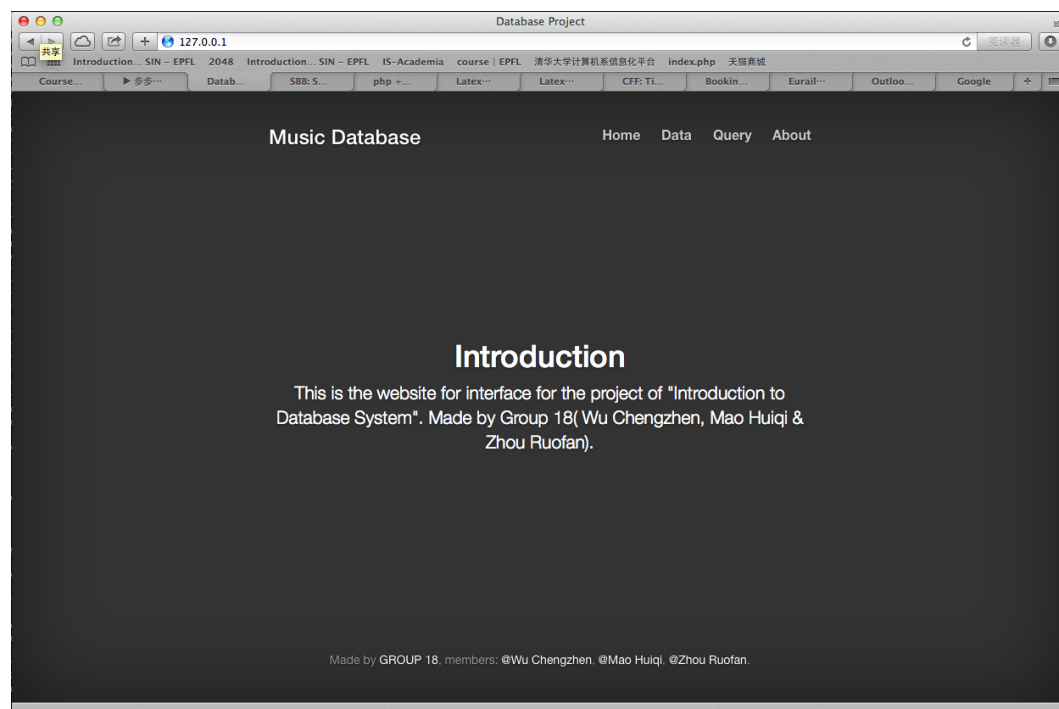
        where track.MID = medium.MID
        group by medium.MID order by count(*) desc;

select med_track.mid, med_track.tracks
      from med_track
     where med_track.tracks =(
            select MAX(med_track.tracks)
            from med_track);

```

6 Interface

Since we've already upload the data to the server, it's convient for us to use PHP + Apache + Oracle to build the website as interface, like below:



It's the index of our website. The website contains 4 parts: Home, Data, Query and About, and the functionally page is Data and Query.

The screen shot of Data page is as below, the page shows the data of tables(you can select the table you want to see by clicking the link of table names, which are the yellow words on the uper part of the page). Each page would show 20 data and you can blowse more data by click the link of pages on the bottom part of the page. As the screen shot, it shows the 'artist' table. By moving your mouse onto those foreign keys, a prompt box

showing the message of the table linked by the foreign key(as the screen shot, we move the mouse onto the 'areaid' and a box showing message of the name and type of the 'areaid').

area	artist	art_genre	art_track	genre	medium	recording	release	track	AREAD
AID	ANAME					ATYPE	GENDER		AREAD
68440	Combo Audio				Group	Other	Other	222	
300542	Matthieu Lagarde				Other	Other	Other	0	City: Petit-Palais-et-Cornemps
318750	The Happies				Other	Other	Other	0	
610232	Lounge Jam				Other	Other	Other	0	
68244	Bazooka Joe				Person	Male	Other	0	
475186	Martina Tichová				Person	Female	Other	56	
147590	The Ulf Sandberg Quartet				Group	Other	Other	0	
163464	Frank Proffitt Jr.				Other	Other	Other	0	
835812	The Other Thing Brass Band				Other	Other	Other	0	
740587	As a People				Other	Other	Other	0	
67048	The Aluminium Group				Other	Other	Other	0	
507683	Patrick HAF				Person	Other	Other	0	
883851	Steffen Horn				Person	Male	Other	160	
68378	Otis Reem				Other	Other	Other	0	
504812	Ben Pollack and His Californians				Group	Other	Other	0	
68380	Megadog				Other	Other	Other	0	
68512	Abfahrt Hinwil				Group	Other	Other	0	
551815	Maxina McClain				Person	Other	Other	0	
68396	Slapdash				Other	Other	Other	0	
162331	Chris Barron				Person	Other	Other	0	

keyword search

[1] 2 3 4 > >>

Made by GROUP 18, members: @Wu Chengzhen, @Mao Huiqi, @Zhou Ruofan.

Under each row there's a input box, and you can use it to search for keyword. Just by clicking the "Keyword Search" you can filter the data of the table. For example, next screen shot shows the result as we input 'Person' under keyword 'atype' and 'male' under keyword 'gender'.

Database Project

127.0.0.1/datas_artist.php?id=&aname=&atype=Person&gender=Male&areaid=&submit=keyword+search

Introduction... SIN - EPFL 2048 Introduction... SIN - EPFL IS-Academia course | EPFL 清华大学计算机系信息化平台 index.php 天猫商城

Course... 步步... Datab... 翻译 php +... Latex-- Latex-- CFF: Ti... Bookin... Eurail... Outloo... Google +

Music Database Home Data Query About

area	artist	art_genre	art_track	genre	medium	recording	release	track
AID	ANAME					ATYPE	GENDER	AREAD
68244	Bazooka Joe					Person	Male	0
883351	Steffen Horn					Person	Male	160
1011628	Elijah Josam					Person	Male	238
423183	Richard Wyands					Person	Male	222
591287	Charles-Émile Beullac					Person	Male	36
655348	El Sais					Person	Male	5187
891262	Παναγιώτης Τσίπορας					Person	Male	84
842193	Silmy Adenoid					Person	Male	0
839173	Osvaldo Berlinger					Person	Male	10
978462	Udo Hüppe					Person	Male	81
978485	Daniel Schweiker					Person	Male	81
978463	Sebastian Bosum					Person	Male	81
978464	Florian Bosum					Person	Male	81
978466	Sebó					Person	Male	81
978491	Martin Stadelmann					Person	Male	203
978469	Piotr Wojtasik					Person	Male	170
978486	José Juan Cadenas					Person	Male	194
978470	Michael Felberbaum					Person	Male	222
978471	Karl Jannuska					Person	Male	38
978500	Christian Käufer					Person	Male	203

keyword search

[1] 2 3 4 > >>

Made by GROUP 18, members: @Wu Chengzhen, @Mao Huiqi, @Zhou Ruofan.

And we satisfied all the queries in the Query page, and you can see the results by clicking the query number.

Database Project

127.0.0.1/query.php

Introduction... SIN - EPFL 2048 Introduction... SIN - EPFL IS-Academia course | EPFL 清华大学计算机系信息化平台 index.php 天猫商城

Course... 步步... Datab... 翻译 php +... Latex-- Latex-- CFF: Ti... Bookin... Eurail... Outloo... Google +

Music Database Home Data Query About

A B C D E F G

Queries: Print the names of artists from Switzerland, i.e., artists whose area is Switzerland.

Combo Audio
Matthieu Lagarde
The Happies
Lounge Jam
Bazooka Joe
Martina Trchová
The Ulf Sandberg Quartet
Frank Proffitt Jr.
The Other Thing Brass
Band
As a People
The Aluminium Group
Patrick HAF
Steffen Horn
Otis Reem
Ben Pollack and His
Californians
Magadog
Abfahrt Hinwil
Maxine McClain
Slapdash
Chris Barron

keyword search

[1] 2 3 4 > >>

Made by GROUP 18, members: @Wu Chengzhen, @Mao Huiqi, @Zhou Ruofan.