Ecole Polytechnique Fédérale de Lausanne

Introduction to Database Systems

Report 1 - Team 04

Maxime Lemarignier (247926)

Yusi Zou (257232)

Olivier Cloux (236079)

March 26th 2018

1 Assumptions

- There exists exactly one biographer field for each existing biography,
- each name we refer to as *Artistname* in the ER model is unique and can be used as a key for the given dataset, even if this is not the case in the real life,
- a clip can have multiple Writers, Producers, Directors and Actor,
- a clip can be of multiple Genres or none,
- a clip can have multiple or no Language,
- a clip can be associated with no country,
- a clip can have no release informations (i.e no releaseCountry)
- a person can have different roles at the same time in a clip (ex : Producer-Director),
- there can be multiple types of links between two clips.

2 Entity Relationship Schema

2.1 Schema

See Figure 1 in appendix A

2.2 Description

2.2.1 Entities

- **Person :** An entity that contains almost all the attributes from the `Biographies` file.
- **Biography :** A weak entity for which each tuple contains a biography, a biographer and the `artistname` of the concerned person, which is set as primary key and references from the `People` entity. The goal of this entity is to improve the memory storage as biographies can be very long strings whereas only few people in the dataset have a complete biography. Therefore we create a biography/biographer tuple only if there exists one.
- **Clips :** An entity used to represent Clips that also contains specific attributes such as their Ratings.
- **Languages :** An entity used to represent Languages. An integer `LangID` is used as primary key since we considered it better practice than to use the language string as a key.

- **Genres :** An entity used to represent Genres. An integer `GenreID` is used as primary key.
- **Countries :** An entity containing Country names that is used to represent both `releaseCountry` and `associatedCountry`.

2.2.2 Relationships

- **PlaysIn, Directs, Writes, Produces :** A person can play in/direct/write/produce one clip or many clips as an actor/director/writer/producer in one or more clips whereas a clip must have at least one actor/director/writer/producer.
- **Linked :** A clip can be linked to another one. The linktype attribute describes the type of link. A clip_from must have a clip_to.
- **HasLang :** A clip can have one or more languages or none.
- **HasGenre :** A clip can have one or more genres or none.
- **Associated :** A clip can be associated to some countries or none.
- **Released :** Depending on the release country, the clip can have different running time or none.

3 Relational Schema

3.1 ER Schema to Relational Schema

- The table *Biography* is separated from *People*, because some of its values exceed the capacity of a CHAR (4000) and have to be stored as a CLOB in Oracle. It may take too much time when we make a request for another information. The primary key used is *ArtistName*, since there can only be one Biography per person and the Biography Entity is a weak entity. Once the person is deleted from the table, the associated biography will also be deleted.
- VARCHAR2 type was used instead of the CHAR type to optimize storage usage, as it accepts a variable size with a limit instead of a fixed length.
- We tried to set the most appropriate maximum length value for each String value int the tables. (ex : 100 chars for the real name).
- We chose to translate each relationship to as a table, as it seemed more natural with our model.

3.2 DDL

```
CREATE TABLE People (
   realname VARCHAR2(100),
   nickname VARCHAR2(100),
   artistname VARCHAR2(100),
   trademark VARCHAR2(100),
   birth VARCHAR2(100),
   death VARCHAR2(100),
   salary INTEGER,
   whereAreTheyNow VARCHAR2(100),
   height VARCHAR2(50),
   spouse VARCHAR2(100),
   biographicalBooks VARCHAR2(1000),
   trivia VARCHAR2(500),
   addInfo VARCHAR2(500),
   personalQuote VARCHAR2(500),
  PRIMARY KEY (artistname)
);

CREATE TABLE Biography(
   biography CLOB,
   biographer VARCHAR2(100),
   artistname VARCHAR2(100),
  PRIMARY KEY (artistname),
  FOREIGN KEY (artistname)
    REFERENCES People
    ON DELETE CASCADE
);

CREATE TABLE Clips(
   clipid INTEGER,
   rank FLOAT,
   cliptitle CHAR(100),
   votes INTEGER,
   clipyear INTEGER,
   cliptype CHAR(2),
  PRIMARY KEY (clipid)
);

CREATE TABLE PlaysIn (
   artistname VARCHAR2(100),
   clipid INTEGER,
   addinfo VARCHAR2(1000),
   chars VARCHAR2(200),
   orderscredits INTEGER,
  PRIMARY KEY (artistname, clipid),
  FOREIGN KEY (clipid)
    REFERENCES Clips,
  FOREIGN KEY (artistname)
    REFERENCES People
);

CREATE TABLE Directs (
   artistname VARCHAR2(100),
   clipid INTEGER,
   addinfo VARCHAR2(1000),
   roles VARCHAR2(200),
  PRIMARY KEY (artistname, clipid),
  FOREIGN KEY (clipid)
    REFERENCES Clips,
  FOREIGN KEY (artistname)
    REFERENCES People
);

CREATE TABLE Produces (
   artistname VARCHAR2(100),
   clipid INTEGER,
   addinfo VARCHAR2(1000),
   roles VARCHAR2(200),
  PRIMARY KEY (artistname, clipid),
  FOREIGN KEY (clipid)
    REFERENCES Clips,
  FOREIGN KEY (artistname)
    REFERENCES People
);

CREATE TABLE Writes (
   artistname VARCHAR2(100),
   clipid INTEGER,
   addinfo VARCHAR2(1000),
   roles VARCHAR2(200),
   worktype VARCHAR2(100),
  PRIMARY KEY (artistname, clipid),
  FOREIGN KEY (clipid)
    REFERENCES Clips,
  FOREIGN KEY (artistname)
    REFERENCES People
);

CREATE TABLE Linked(
   clipto INTEGER,
   clipfrom INTEGER,
   linktype VARCHAR2(50),
  PRIMARY KEY (clipto,clipfrom,linktype),
  FOREIGN KEY (clipto)
    REFERENCES Clips,
  FOREIGN KEY (clipfrom)
    REFERENCES Clips
);

CREATE TABLE Languages(
   langid INTEGER,
   language VARCHAR2(50),
  PRIMARY KEY (langid)
);

CREATE TABLE HasLang(
   clipid INTEGER,
   langid INTEGER,
  PRIMARY KEY (clipid, langid),
  FOREIGN KEY (clipid)
    REFERENCES Clips,
  FOREIGN KEY (langid)
    REFERENCES Languages
);

CREATE TABLE Genres(
   genreid INTEGER,
   genre VARCHAR2(20),
  PRIMARY KEY (genreid)
);

CREATE TABLE HasGenre(
   clipid INTEGER,
   genreid INTEGER,
  PRIMARY KEY (clipid, genreid),
  FOREIGN KEY (clipid)
    REFERENCES Clips,
  FOREIGN KEY (genreid)
    REFERENCES Genres
);

CREATE TABLE Countries(
   countryid INTEGER,
   country VARCHAR2(50),
  PRIMARY KEY (countryid)
);

CREATE TABLE Associated (
   clipid INTEGER,
   countryid INTEGER,
  PRIMARY KEY (clipid, countryid),
  FOREIGN KEY (clipid)
    REFERENCES Clips,
  FOREIGN KEY (countryid)
    REFERENCES Countries
);

CREATE TABLE Released (
   clipid INTEGER,
   countryid INTEGER,
   releasedate VARCHAR2(20),
   runningtime INTEGER,
  PRIMARY KEY (clipid, countryid),
  FOREIGN KEY (clipid)
    REFERENCES Clips,
  FOREIGN KEY (countryid)
    REFERENCES Countries
);
```

4 General Comments

Every team member was involved in the making of this assignment. It helped us to get a better understanding of the lecture. The provided data are not clean and they were some difficulties to well implement the structure, but it may happen in real life.

A Figures

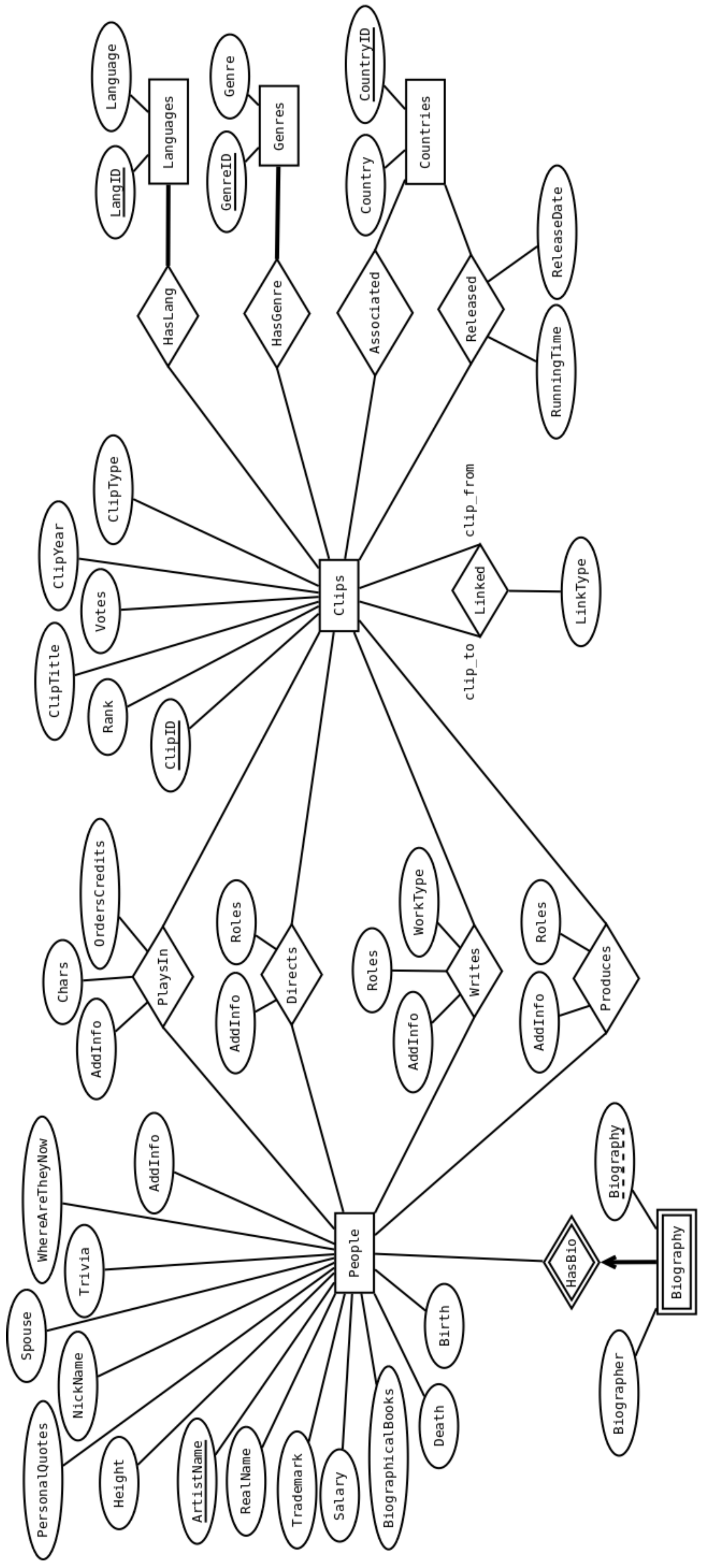

Figure 1: Our model