École Polytechnique Fédérale de Lausanne

Databases Project
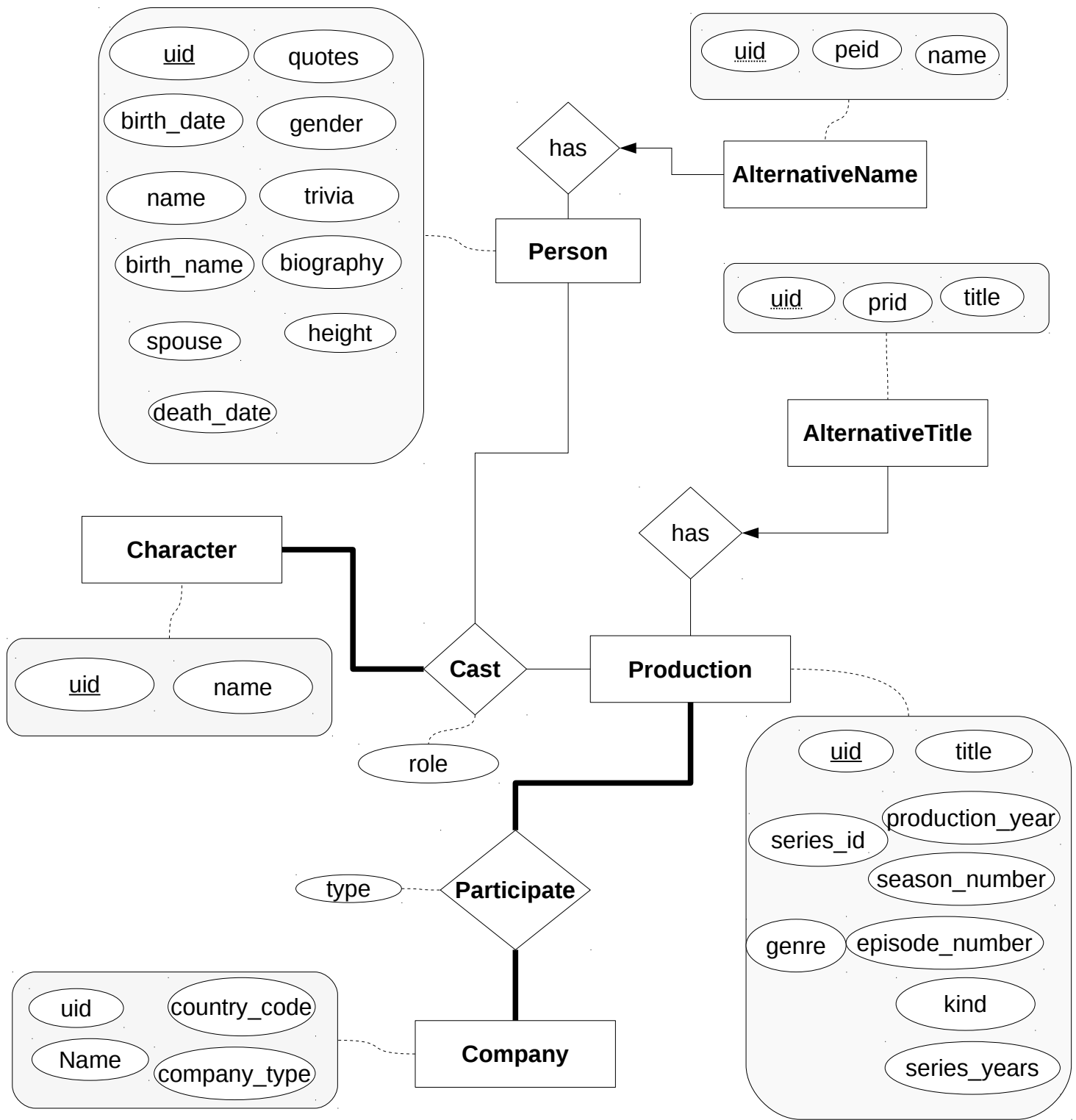
First deliverable

Authors :
Tristan Overney
Morgan Bruhin
Valentine Arrieta

22 mars 2015

1 ER model

Person
- uid
- quotes
- birth_date
- gender
- name
- trivia
- birth_name
- biography
- spouse
- height
- death_date

AlternativeName
- uid
- peid
- name

AlternativeTitle
- uid
- prid
- title

Person —— has ——◁—— **AlternativeName**

AlternativeTitle —— has ——◁—— **Production**

Character
- uid
- name

Character —— **Cast** —— **Production**

- role

Production
- uid
- title
- series_id
- production_year
- season_number
- genre
- episode_number
- kind
- series_years

Cast —— **Participate** —— **Company**

- type

Company
- uid
- country_code
- Name
- company_type

2 Table creation

We will implement the next deliverables using our own server with PostGreSQL. As such the types used in the following CREATE TABLE possibly doesn't work well in an Oracle DB environnement.

```
1  CREATE TABLE Person
2    (uid INTEGER NOT NULL ,
3     first_name CHAR (75) NOT NULL ,
4     last_name CHAR (75) NOT NULL ,
5     gender CHAR (1) ,
6     trivia TEXT ,
7     quotes VARCHAR (4000) ,
8     birth DATE ,
9     death DATE ,
10    biography TEXT ,
11    spouse CHAR (100) ,
12    height DOUBLE PRECISION ,
13    primary key (uid));
14
15 CREATE TABLE AlternativeName
16    (uid INTEGER NOT NULL ,
17     pid INTEGER NOT NULL ,
18     name CHAR (60) NOT NULL ,
19     primary key (uid)
20     foreign key (pid) references Person(uid),
21     ON DELETE CASCADE);
22
23 CREATE TABLE Character
24    (uid INTEGER NOT NULL ,
25     name CHAR (60) NOT NULL ,
26     primary key (uid),
27     );
28
29 CREATE TABLE Cast
30    (cid INTEGER ,
31     perid INTEGER NOT NULL ,
32     prodid INTEGER NOT NULL ,
33     rid SMALLINT NOT NULL ,
34     primary key (cid, perid, prodid , role),
35     foreign key (cid) references Character (uid),
36     foreign key (perid) references Person (uid),
37     foreign key (prodid) references Production (uid)
38     foreign key (rid) references Role (uid));
39
40 CREATE TABLE Role
41    (uid SMALLINT NOT NULL ,
42     title CHAR (20) NOT NULL ,
43     primary key (uid));
44
45 CREATE TABLE Production
46    (uid INTEGER NOT NULL ,
47     title CHAR (80) NOT NULL ,
48     currency CHAR (3) ,
49     budget INTEGER ,
50     primary key (uid));
```

```
51
52  CREATE TABLE Participate
53      (pid INTEGER NOT NULL ,
54       cid INTEGER NOT NULL ,
55       type CHAR (20) NOT NULL ,
56       primary key (pid , cid)
57       foreign key (pid) references Production (uid) ,
58       foreign key (cid) references Company (uid));
59
60  CREATE TABLE Company
61      (uid INTEGER NOT NULL ,
62       name CHAR (20) NOT NULL ,
63       country_code CHAR (6) ,
64       primary key (uid));
65
66  CREATE TABLE AlternativeTitle
67      (uid INTEGER NOT NULL ,
68       pid INTEGER ,
69       title CHAR (30) ,
70       primary key (uid , pid)
71       foreign key (pid) references Production (uid) ,
72       ON DELETE CASCADE);
```

3 Discussion about constraint

A person can have one or more alternative name but an alternative name can describe only one person. We thus have a one-to-many relationship set. Also if the person is deleted from the database, his alternative names (if any) will be too. We thus created a weak entity for alternative name.

Same thing for the production that can have one or more alternative title, we created a weak entity for alternative title.

A production cast make the connection between the production, the person and the character (if any). We thus made a ternary relationship between the three entity person, character and production. We add the role of the person as an attribute of the relationship set. A character exist only if related to at least one person and one production. The person and production entity don't have this constraint since a production can have only a director that won't have any character.

Companies can be involved into production. They can play roles of different type (for example production, distribution), thus we added an attribute to the relationship "participate" called type (since the data description was changed in that way before the deadline this justification isn't relevant any more). A production need to have at least one company involved.