

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE



## Databases Project

---

*First deliverable*

*Authors :*

Tristan OVERNEY  
Morgan BRUHIN  
Valentine ARRIETA

22 mars 2015

# 1 ER model

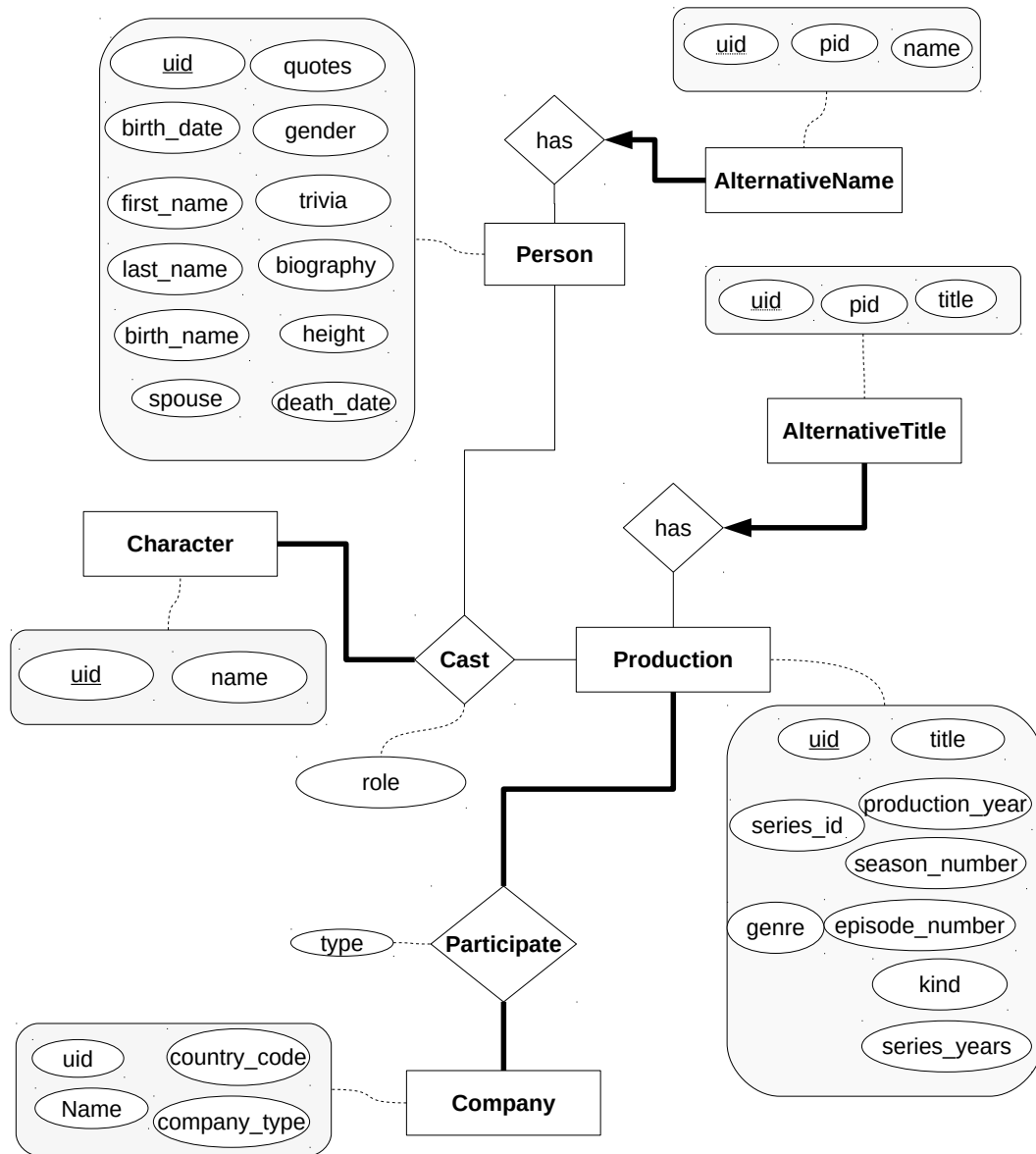


FIGURE 1 – ER model of our DBMS

## 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.

```
CREATE TABLE Person
(uid INTEGER NOT NULL,
 first_name CHAR(75) NOT NULL,
 last_name CHAR(75) NOT NULL,
 gender CHAR(1),
 trivia TEXT,
 quotes VARCHAR(4000),
 birth DATE,
 death DATE,
 biography TEXT,
 spouse CHAR(100),
 height DOUBLE PRECISION,
 primary key (uid));

CREATE TABLE AlternativeName
(uid INTEGER NOT NULL,
 pid INTEGER NOT NULL,
 name CHAR(60) NOT NULL,
 primary key (uid)
 foreign key (pid) references Person(uid),
 ON DELETE CASCADE);

CREATE TABLE Character
(uid INTEGER NOT NULL,
 name CHAR(60) NOT NULL,
 primary key (uid),
 );

CREATE TYPE CAST_ROLE AS
ENUM ('actor', 'actress', 'producer', 'writer', 'cinematographer',
 'composer', 'costume_designer', 'director', 'editor', 'miscellaneous_crew',
 'production_designer');

CREATE TABLE Cast
(cid INTEGER,
 perid INTEGER NOT NULL,
 prodid INTEGER NOT NULL,
 rid CAST_ROLE NOT NULL,
 primary key (cid, perid, prodid, role),
 foreign key (cid) references Character (uid),
 foreign key (perid) references Person (uid),
 foreign key (prodid) references Production (uid)
 foreign key (rid) references Role (uid));

CREATE TYPE PROD_KIND AS
ENUM ('tv_series', 'movie', 'episode', 'movie', 'video_movie', 'video_game');

CREATE TABLE Production
(uid INTEGER NOT NULL,
 title CHAR(80) NOT NULL,
```

```

        production_year DATE,
        sid INTEGER,
        season SMALLINT,
        episode SMALLINT,
        series_years CHAR (11),
        kind PROD_KIND NOT NULL,
        genre CHAR(20),
        primary key (uid));

CREATE TYPE COMPANY_TYPE AS ENUM ('distributors', 'production_company');

CREATE TABLE Participate
(pid INTEGER NOT NULL,
cid INTEGER NOT NULL,
type COMPANY_TYPE NOT NULL,
primary key (pid, cid)
foreign key (pid) references Production(uid),
foreign key (cid) references Company(uid));

CREATE TABLE Company
(uid INTEGER NOT NULL,
name CHAR(80) NOT NULL,
country_code CHAR(6),
primary key (uid));

CREATE TABLE AlternativeTitle
(uid INTEGER NOT NULL,
pid INTEGER NOT NULL,
title CHAR (80) NOT NULL,
primary key (uid, pid)
foreign key (pid) references Production (uid),
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 instead of having a lot of duplicated entries in the "Company" table. Since that design choice, the data set given has been changed and comforted us in our design choice as the "type" attribute has been moved out of the COMPANY.csv file. A production need to have at least one company involved.