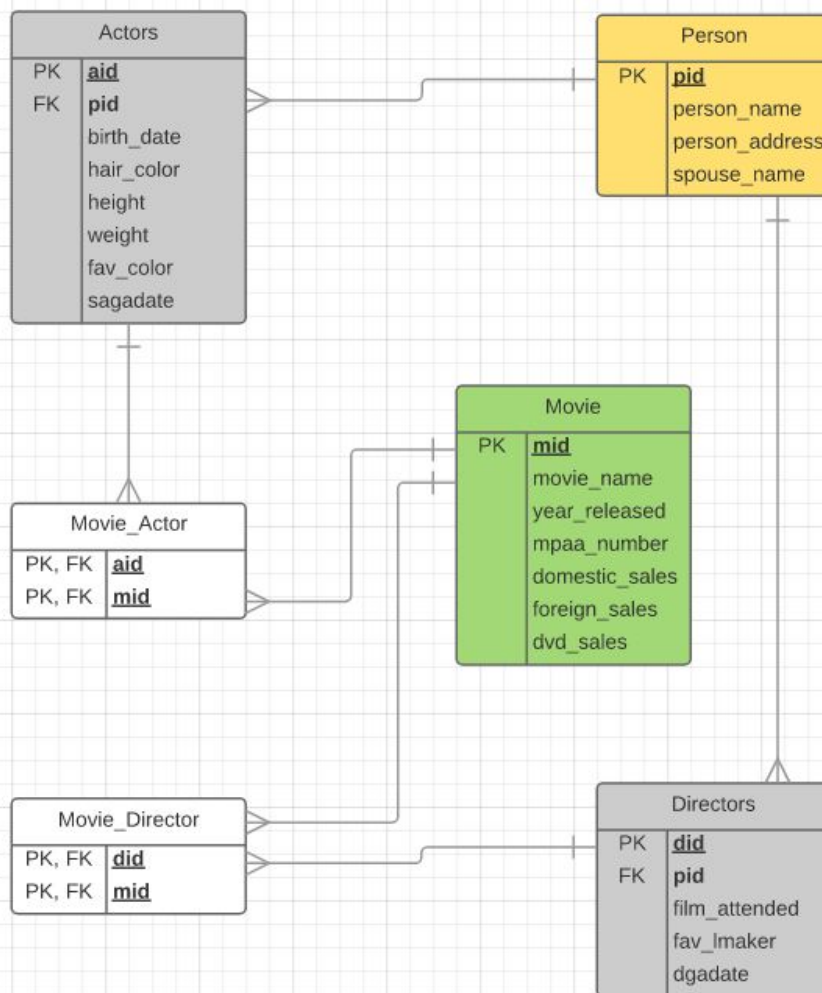


- Database: "Mixar"
- Masson Celestin
- Database Management
- Professor: Pablo Rivas
- Lab 8

Célestin Masson | April 13, 2016



DROP DATABASE IF EXISTS "Mixar";

CREATE DATABASE "Mixar";

/* Functional dependencies

ACTORS

Aid > birth_date, hair_color, height, weight, fav_color, sagadate, pid

MOVIE

Mid > movie_name, year_released, mpaa_number, domestic_sales, foreign_sales,
dvd_sales

MOVIE ACTOR

(Aid, mid) > aid, mid

MOVIE DIRECTOR

(did, mid) > did, mid

DIRECTOR

Did > film_attended, dgadate, fav_lmaker, pid

PERSON

Pid > person_name, person_address, spouse_name

***/**

CREATE TABLE IF NOT EXISTS Actors (
 Aid INTEGER UNIQUE NOT NULL,
 Pid INTEGER NOT NULL,
 Birth_date DATE NOT NULL,

```
    Hair_color VARCHAR(3),
    Height INTEGER,
    Weight INTEGER,
    Fav_color VARCHAR(3),
    Sagadate DATE,
    PRIMARY KEY (Aid),
    FOREIGN KEY (pid) REFERENCES person(pid)
);
```

```
CREATE TABLE IF NOT EXISTS Movie (
    Mid INTEGER UNIQUE NOT NULL,
    Movie_name VARCHAR(100) NOT NULL,
    Year_released INTEGER NOT NULL,
    Mpaa_number INTEGER,
    Domestic_sales INTEGER,
    Foreign_sales INTEGER,
    Dvd_sales INTEGER,
    PRIMARY KEY (mid)
);
```

```
CREATE TABLE IF NOT EXISTS Director (
    Did INTEGER UNIQUE NOT NULL,
    Film_attended VARCHAR(100),
    Dgadata DATE,
    Fav_lmaked VARCHAR(100),
    Pid INTEGER NOT NULL,
    PRIMARY KEY (did),
    FOREIGN KEY (pid) REFERENCES person(pid)
);
```

```
CREATE TABLE IF NOT EXISTS Movie_Actor(
    Aid INTEGER NOT NULL,
    Mid INTEGER NOT NULL,
    PRIMARY KEY (Aid, Mid),
    FOREIGN KEY (aid) REFERENCES Actors(aid),
    FOREIGN KEY (mid) REFERENCES Movie(Mid)
);
```

```
CREATE TABLE IF NOT EXISTS Movie_Director(
    Did INTEGER NOT NULL,
    Mid INTEGER NOT NULL,
    PRIMARY KEY (did, mid),
    FOREIGN KEY (did) REFERENCES Director(did),
    FOREIGN KEY (mid) REFERENCES Movie(Mid)
```

```
);
```

```
CREATE TABLE IF NOT EXISTS Person(  
    Pid INTEGER UNIQUE NOT NULL,  
    Person_name VARCHAR(50) NOT NULL,  
    Person_address VARCHAR(150) NOT NULL,  
    Spouse_name VARCHAR(50),  
    PRIMARY KEY (pid)  
);
```

```
INSERT INTO Person(pid, person_name, person_address, spouse_name)  
VALUES (1, 'Keanu Reeves', '123', 'spouse');
```

```
INSERT INTO Actors(aid, pid, birth_date)  
VALUES (1, 1, '1/1/11');
```

```
INSERT INTO Person(pid, person_name, person_address, spouse_name)  
VALUES (2, 'Steven Spielberg', '321', 'spouse2');
```

```
INSERT INTO Director(did, pid)  
VALUES (1, 2);
```

```
INSERT INTO Movie(mid, movie_name, year_released)  
VALUES (1, 'Mickey Mouse', 1999);
```

```
SELECT * FROM Movie;
```

```
INSERT INTO Movie_Actor(aid, mid)  
VALUES (1,1);
```

```
INSERT INTO Movie_Director(did, mid)  
VALUES (1,1);
```

```
SELECT * FROM Person;
```

```
SELECT person_name  
FROM person  
WHERE pid IN (  
    SELECT pid  
    FROM director  
    WHERE did IN (  
        SELECT did  
        FROM movie_director  
        WHERE mid IN (  
            SELECT mid  
            FROM movie_actor  
            WHERE aid = 1  
        )  
    )  
);
```

```
SELECT mid
FROM movie_actor
WHERE aid IN (
    SELECT aid
    FROM actors
    WHERE pid = (
        SELECT pid
        FROM Person
        WHERE person_name = 'Keanu Reeves'))))));
```