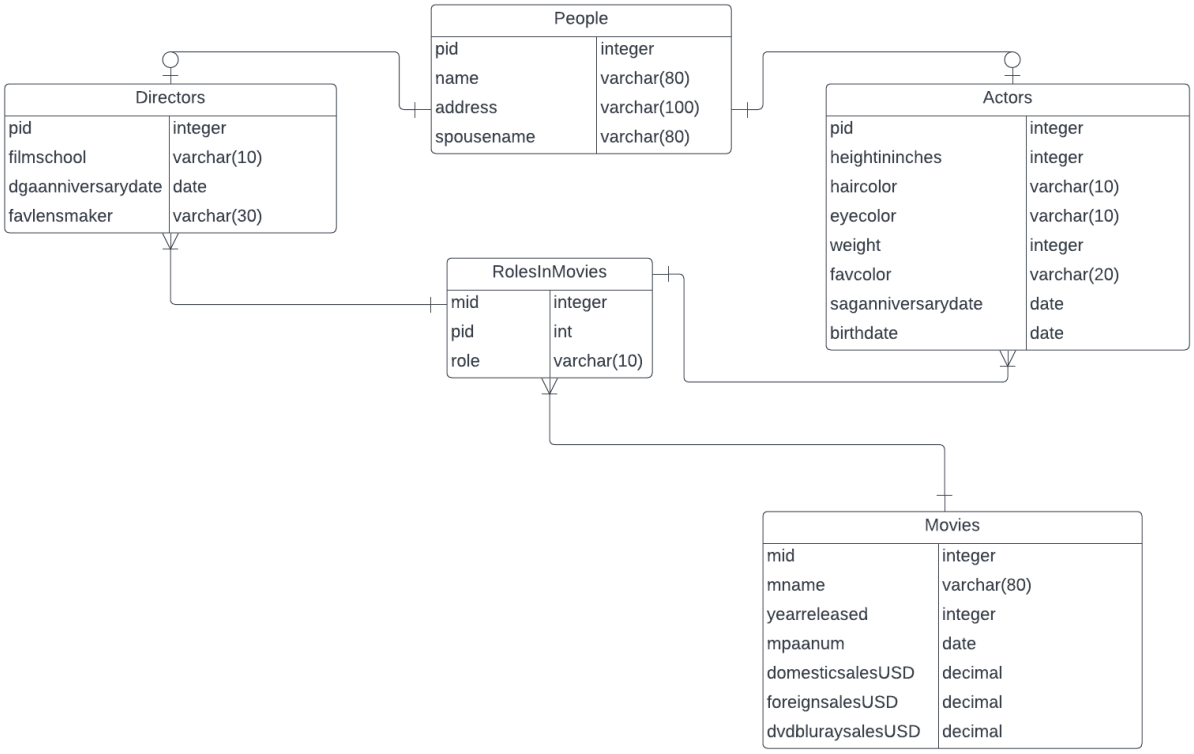
Raymond Tomo
04/13/22

Lab 08: Normalization

E/R Diagram



SQL Create Statements

People

Established key for people in the database

```
CREATE TABLE People(
        pid       INTEGER UNIQUE NOT NULL,
        name      VARCHAR (80)  NOT NULL,
        address   VARCHAR (100) NOT NULL,
        spousename VARCHAR (80),
  PRIMARY KEY (pid)
);
```

Actors

Established subtype for People who are actors)

```
CREATE TABLE Actors(
        pid       INTEGER UNIQUE NOT NULL REFERENCES People(pid),
        heightininches      VARCHAR (80)  NOT NULL,
        haircolor  VARCHAR (10) NOT NULL,
        eyecolor  VARCHAR (10) NOT NULL,
        weight       INTEGER NOT NULL,
        favcolor  VARCHAR (10) NOT NULL,
        saganniversarydate DATE NOT NULL,
        birthdate DATE NOT NULL,
   PRIMARY KEY(pid)
);
```

Directors

Established subtype for People who are directors

```
CREATE TABLE Directors(
        pid       INTEGER UNIQUE NOT NULL REFERENCES People(pid),
        filmschool  VARCHAR (10) NOT NULL,
        dgaanniversarydate DATE NOT NULL,
        favlensmaker  VARCHAR (30) NOT NULL,
   PRIMARY KEY(pid)
);
```

Movies

Creates key for movies that can be referenced in the table

```
CREATE TABLE Movies(
        mid       INTEGER UNIQUE NOT NULL,
        mname VARCHAR(80) NOT NULL,
        yearreleased     INTEGER NOT NULL,
        mpaanum  INTEGER NOT NULL,
        domesticsalesUSD DECIMAL NOT NULL,
        foreignsalesUSD DECIMAL NOT NULL,
        dvdbluraysalesUSD DECIMAL  NOT NULL,
   PRIMARY KEY(mid)
);
```

RolesInMovies

Links both movies and people together using a combination key(similar to Lab 7), uses role to separate actors from directors in case someone is an actor in one film and a director in another.

```
CREATE TABLE RolesInMovies(
        pid    INTEGER UNIQUE NOT NULL REFERENCES People(pid),
```

```
        mid      INTEGER UNIQUE NOT NULL REFERENCES Movies(mid),
        role  VARCHAR (10) NOT NULL,
  PRIMARY KEY (pid,mid)
);
```

Functional Dependencies

People

- PID -> name, address, spousename

Actors

- PID ->heightininches,haircolor,eyecolor, weight,favcolor,sgaaanniversarydate,birthdate

Directors

- PID -> filmschool, dgaanniversaydate, favlensmaker

RolesInMovies

- PID, MID -> role

Movies

- MID -> mname,yearrelased, mpaanum, domesticsalesUSD, foreignsalesUSD, dvdblurayUSD

SQL Query

```
SELECT DISTINCT People.pid, People.name, People.address, People.spousename,
Directors.filmschool, Directors.dgaanniversarydate, Directors favlensmaker
FROM People, Directors
WHERE Directors.pid in(SELECT RolesInMovies.pid
                       FROM RolesInMovies
                       WHERE mid in (SELECT RolesInMovies.mid
                                     FROM RolesInMovies
                                     WHERE pid in
                                     (SELECT pid
                                      FROM  People
                                      WHERE name = “Roger Moore”)));
```