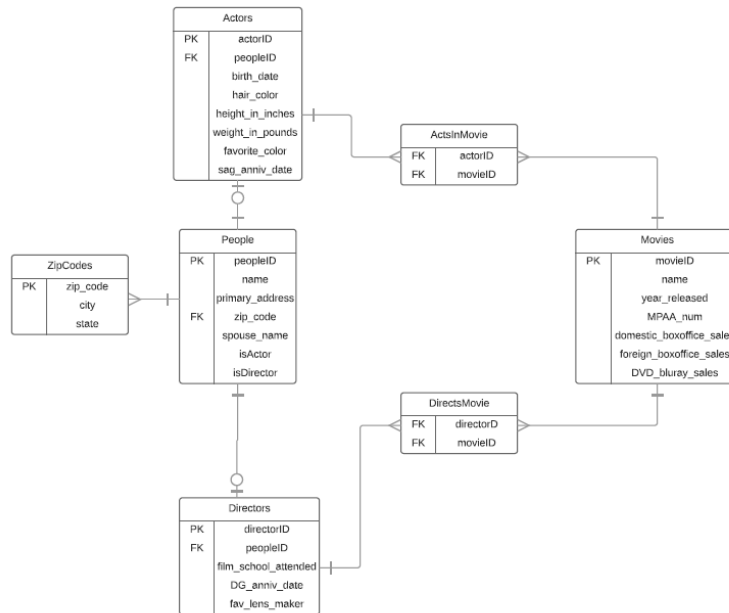Jillian Preece
CMPT 308
March 27, 2017

1.

2.

DROP TABLE IF EXISTS Actors;
DROP TABLE IF EXISTS Directors;
DROP TABLE IF EXISTS Movies;
DROP TABLE IF EXISTS People;
DROP TABLE IF EXISTS ZipCodes;
DROP TABLE IF EXISTS ActsInMovie;
DROP TABLE IF EXISTS DirectsMovie;

--People--
CREATE TABLE People (
    peopleID   char(5) not null,
    name   text,
    primary_table   text,
    zip_code   char(5) not null references ZipCodes(zip_code),
    spouse_name   text,
    isActor   boolean,
    isDirector   boolean,
primary key(peopleID)
);

```sql
-- Actors --
CREATE TABLE Actors (
 actorID          char(5) not null,
 peopleID         char(5) not null references People(peopleID),
 birth_date       date,
 hair_color       text,
 eye_color        text,
 height_in_inches    integer,
 weight_in_lbs       integer,
 favorite_color      text,
 SAG_anniv_date      date,
 primary key(peopleID)
);

-- Directors --
CREATE TABLE Directors (
 directorID        char(5) not null,
 peopleID                          char(5) not null references People(peopleID),
 film_school_attended  text,
 DG_anniv_date       date,
 fave_lens_maker      text,
 primary key(did)
);

-- Movies --
CREATE TABLE Movies (
 movieID             char(3) not null,
 name                text,
 year_released         char(4),
 MPAA_num              integer,
 domesic_boxoffice_sales   numeric(10,2),
 foreign_boxoffice_sales   numeric(10,2),
 DVD_bluray_sale         numeric(10,2),
 primary key(movieID)
);

-- Zip Codes --
CREATE TABLE ZipCodes (
 zip_code    char(5) not null,
 city        text,
 state       text,
primary key(zip_code)
);
```

```
-- ActsInMovie --
CREATE TABLE ActsInMovie (
  actorID    char(5) not null references Actors(actorID),
  movieID        char(5) not null references movies(movieID),
primary key(peopleID, movieID)
);

-- DirectsMovie --
CREATE TABLE DirectsMovie (
  directorID      char(5) not null references Directors(directorID),
  movieID    char(5) not null references movies(movieID),
primary key(peopleID, movieID)
);
```

3.
People : peopleID → name, primary_address, zip_code, spouse_name, isActor, isDirector
Actors: peopleID → actorID
        actorID → birth_date, hair_color, height_in_inches, weight_in_pounds, favorite_color,
sag_anniv_date
Directors: peopleID → directorID
            directorID → film_school_attended, DG_anniv_date, fav_lens_maker

ZipCodes: zip_code → city, state

Movies: movieID → name, year_released, MPAA_num, domestic_boxoffice_sales,
foreign_boxoffice_sales, DVD_bluray_sales

ActsInMovie: movieID →actorID

DirectsMovie: movieID → directorID

4.

```
SELECT *
FROM People
WHERE peopleID IN (
                SELECT peopleID
                FROM Directors
                WHERE directorID (
                                SELECT directorID
                                FROM DirectsMovies
                                WHERE movieID IN (
                                                SELECT movieID
                                                FROM ActsInMovie
                                                WHERE actorID IN (
                                                                SELECT peopleID
```

```
                         FROM actors
                         WHERE peopleID in (
                                        SELECT peopleID
                                        FROM People
                                        WHERE name = 'Sean Connery'
                                            )
                                    )
                              )
                         )
                    )
              )
```