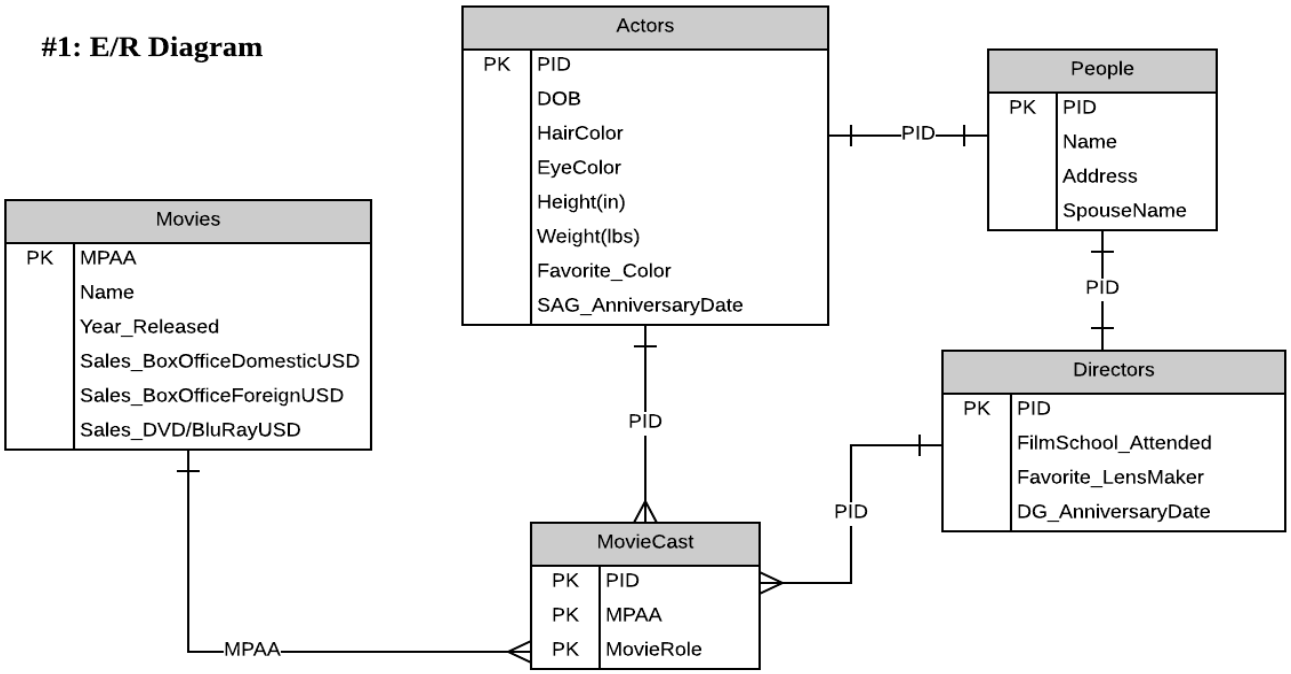
#1: E/R Diagram



2: SQL CREATE STATEMENTS FOR EACH TABLE

```
DROP TABLE IF EXISTS MovieCast;
DROP TABLE IF EXISTS Directors;
DROP TABLE IF EXISTS Actors;
DROP TABLE IF EXISTS People;
DROP TABLE IF EXISTS Movies;

CREATE TABLE People (
  PID char(5) NOT NULL,
  Name text NOT NULL,
  Address text,
  SpouseName text,
  PRIMARY KEY (PID)
);

CREATE TABLE Actors (
  PID char(5) NOT NULL references People(PID),
  DOB date,
  HairColor text,
  EyeColor text,
  HeightIN numeric(5,2),
  WeightLBS numeric(5,2),
  Favorite_Color text,
```

```
  SAG_AnniversaryDate date,
  PRIMARY KEY (PID)
);

CREATE TABLE Movies (
  MPAA int NOT NULL,
  Name text,
  Year_Released int,
  Sales_BoxOfficeDomesticUSD text,
  Sales_BoxOfficeForeignUSD text,
  Sales_DVD_BluRayUSD text,
  PRIMARY KEY (MPAA)
);

CREATE TABLE MovieCast (
  PID char(5) NOT NULL references People(PID),
  MovieRole text NOT NULL,
  MPAA int references Movies(MPAA),
  PRIMARY KEY (PID, MovieRole, MPAA)
);

CREATE TABLE Directors (
  PID char(5) NOT NULL references People(PID),
  FilmSchool_Attended text,
  Favorite_LensMaker text,
  DG_AnniversaryDate text,
  PRIMARY KEY (PID)
);
```

3. FUNCTIONAL DEPENDENCIES FOR EACH TABLE

People: PID → Name, Address, SpouseName

Actors: PID → DOB, HairColor, EyeColor, HeightIN, WeightLBS, Favorite_Color, SAG_AnniversayDate

Movies: MPAA → Name, Year_Released, Sales_BoxOfficeDomesticUSD, Sales_BoxOfficeForeignUSD, Sales_DVD_BluRayUSD,

MovieCast: PID, MovieRole, MPAA →

Directors: PID → FilmSchool_Attended, Favorite_LensMaker, DG_AnninersaryDate

4. WRITE A QUERY TO SHOW ALL THE DIRECTORS WITH WHOM ACTOR "SEAN CONNERY" HAS WORKED.

```
SELECT people.name
FROM people INNER JOIN moviecast ON people.pid = moviecast.pid
WHERE movierole = 'Director'
AND moviecast.mpaa IN ( SELECT moviecast.mpaa
                FROM moviecast INNER JOIN people ON people.pid = moviecast.pid
                WHERE people.name = 'Sean Connery' ) ;
```