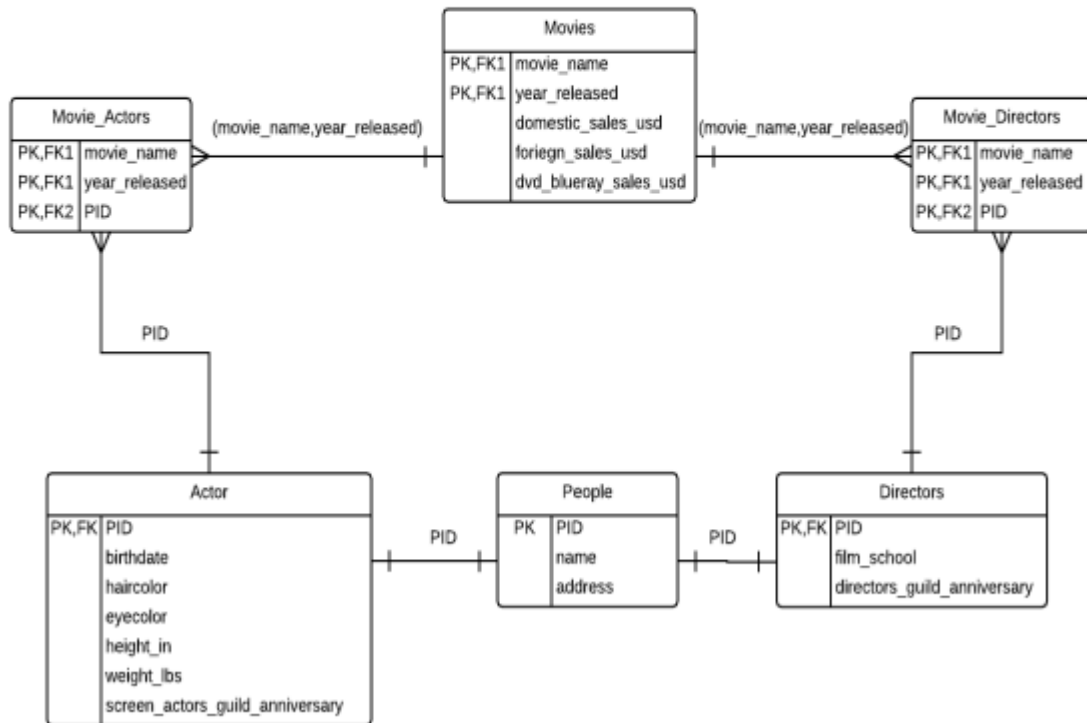


Lab 08

1.)

Lab 8 Movies Entity Relationship Diagram



2.)

```

DROP TABLE IF EXISTS movie_actors;
DROP TABLE IF EXISTS movie_directors;
DROP TABLE IF EXISTS directors;
DROP TABLE IF EXISTS actors;
DROP TABLE IF EXISTS people;
DROP TABLE IF EXISTS movies;
    
```

```

CREATE TABLE Movies (
    movie_name      text not null,
    year_released   smallint CHECK(year_released > 1800),
    domestic_sales_usd numeric(12,2),
    foreign_sales_usd numeric(12,2),
    
```

```
    dvd_blueray_sales_usd    numeric(12,2),  
    PRIMARY KEY(movie_name,year_released)  
);
```

```
CREATE TABLE People(  
    pid                        char(6) not null,  
    name                      text not null,  
    address                  text,  
    UNIQUE(name,address),  
    PRIMARY KEY(pid)  
);
```

```
CREATE TABLE Actors (  
    pid                      char(6) not null REFERENCES People(pid),  
    birthdate               date,  
    eyecolor                char(15),  
    haircolor               char(15),  
    height_in               smallint CHECK(height_in > 0),  
    weight_lbs              smallint CHECK(weight_lbs > 0),  
    screen_actors_guild_anniversary date,  
    PRIMARY KEY(pid)  
);
```

```
CREATE TABLE Directors(  
    pid                      char(6) not null REFERENCES People(pid),  
    film_school              text,  
    directors_guild_anniversary date,  
    PRIMARY KEY(pid)  
);
```

```
CREATE TABLE Movie_Actors(  
    movie_name              text not null,  
    year_released           smallint not null,  
    pid                     char(6) not null REFERENCES Actors(pid),  
    FOREIGN KEY(movie_name,year_released) REFERENCES  
        Movies(movie_name,year_released),  
    PRIMARY KEY(movie_name,year_released,pid)  
);
```

```
CREATE TABLE Movie_Directors(  
    movie_name              text not null,  
    year_released           smallint not null,  
    pid                     char(6) not null REFERENCES Directors(pid),
```

```
FOREIGN KEY(movie_name,year_released) REFERENCES
  Movies(movie_name,year_released),
PRIMARY KEY(movie_name,year_released,pid)
);
```

3.)

```
INSERT INTO
Movies(movie_name,year_released,domestic_sales_usd,foriegn_sales_usd,dvd_blueray_s
ales_usd)
VALUES('Entrapment',1999,87704396.00,124700000.00, 1500000.00),
('Finding Forrester',2000, 51768623.00 , 28245000.00,1245555.00 ),
('The League of Extraordinary Gentlemen',2003, 66465204.00, 112800000.00,
36400000.00 ),
('GoldenEye',1995,106429941.00,245764093.00,22389000.00);
```

```
INSERT INTO People(pid,name,address)
VALUES('p1','Sean Connery','Fake Address 3033, Podunk, NW 32113'),
('p2','Catherine Zeta-Jones','2108 Elm Street, Ellenville, NY 10482'),
('p3','Rob Brown','6792 Pine Road, Trenton, NJ 39802'),
('p4','Stuart Townsend','1293 Empire Ave, Taconic, NY 31123'),
('p5','Pierce Brosnan','9321 High Street, Pheonix, AZ 11234'),
('p6','Jon Amiel','8203 Underwood Road, Endstown, WZ 32393'),
('p7','Gus Van Sant','Fake Address 3033, Podunk, NW 32113'),
('p8','Stephen Norrington','3829 Infinite Ion Ave., Nowville, TX 00789'),
('p9','Martin Campbell','9987 End Of Line Road, Final Town, GA 23225');
```

```
INSERT INTO
Actors(pid,birthdate,haircolor,eyecolor,height_in,weight_lbs,screen_actors_guild_anniver
sary)
VALUES('p1', '08-25-1930','brown', 'brown', 56, 216,'06-18-1954' ),
('p2','07-12-1976','blond','blue',60,120,'07-01-1995'),
('p3','11-25-1980','black','green',67,140,'02-03-1990'),
('p4','02-06-1984','red','hazel',53,210,'04-07-1997'),
('p5','05-21-1970','brown','red',57,175,'08-21-1982');
```

```
INSERT INTO Directors(pid,film_school,directors_guild_anniversary)
VALUES('p6','University of Cambridge','03-24-1982'),
('p7','Film School Of Nowhereville','11-11-1990'),
('p8','Bright Eyes School Of Movies','09-01-1995'),
('p9','The World Of Film','12-30-1987');
```

```
INSERT INTO Movie_Actors(movie_name,year_released,pid)
VALUES('Entrapment',1999,'p1'),
```

```
('Entrapment',1999,'p2'),  
('Finding Forrester',2000,'p1'),  
('Finding Forrester',2000,'p3'),  
('The League of Extraordinary Gentlemen',2003,'p1'),  
('The League of Extraordinary Gentlemen',2003,'p4'),  
('GoldenEye',1995,'p5');
```

```
INSERT INTO Movie_Directors(movie_name,year_released,pid)  
VALUES('Entrapment',1999,'p6'),  
('Finding Forrester',2000,'p7'),  
('The League of Extraordinary Gentlemen',2003,'p8'),  
('GoldenEye',1995,'p9');
```

4.) Movies

(movie_name, year_released) -> the set and all subsets of {domestic_sales_usd,
foreign_sales_usd, dvd_blueray_sales_dvd}
* If I created an artificial key of movie id, it would not be in BCNF. *

People

pid -> the set and all subsets of {name, address}
* at first I thought in theory there could be two "Hosea Rodriguez" at the
same address, but in reality this is probably not the case. To put into
BCNF, I eliminated the transient dependency of (name,address)->pid
with a unique constraint on (name,address)*

Actors

pid -> the set and all subsets of {birthdate, haircolor, eyecolor, height_in,
weight_lbs, screen_actors_guild_anniversary}

Directors

pid-> the set and all subsets of {film_school, directors_guild_anniversary}

Movie Actors

(movie_name,year_released,pid) -> * None *

Movie Directors

(movie_name,year_released,pid) -> * None *

5.)

```
select distinct DirectorsJoin.name as "Directors Name" from movies  
inner join movie_directors on movie_directors.movie_name = movies.movie_name  
AND movie_directors.year_released = movie_directors.year_released  
inner join movie_actors on movie_actors.movie_name = movies.movie_name AND  
movie_actors.year_released = movie_actors.year_released  
inner join people as ActorsJoin on ActorsJoin.pid = movie_actors.pid  
inner join people as DirectorsJoin on DirectorsJoin.pid = movie_directors.pid
```

```
where ActorsJoin.name = 'Sean Connery'  
order by DirectorsJoin.name asc;
```