Normalization HW 2

1. ER diagram

People	
PK	peopleID
	firstName
	lastName
	address

Actors	
PK, FK	actorID
	birthDate
	hairColor
	eyeColor
	heightInches
	weightLb
	guildAnniversary

Directors	
PK, FK	directorID
	filmSchoolAttended
	guildAnniversary

MoviesActed	
PK, FK	movieID
PK, FK	actorID

MoviesDirected	
PK, FK	movieID
PK, FK	directorID

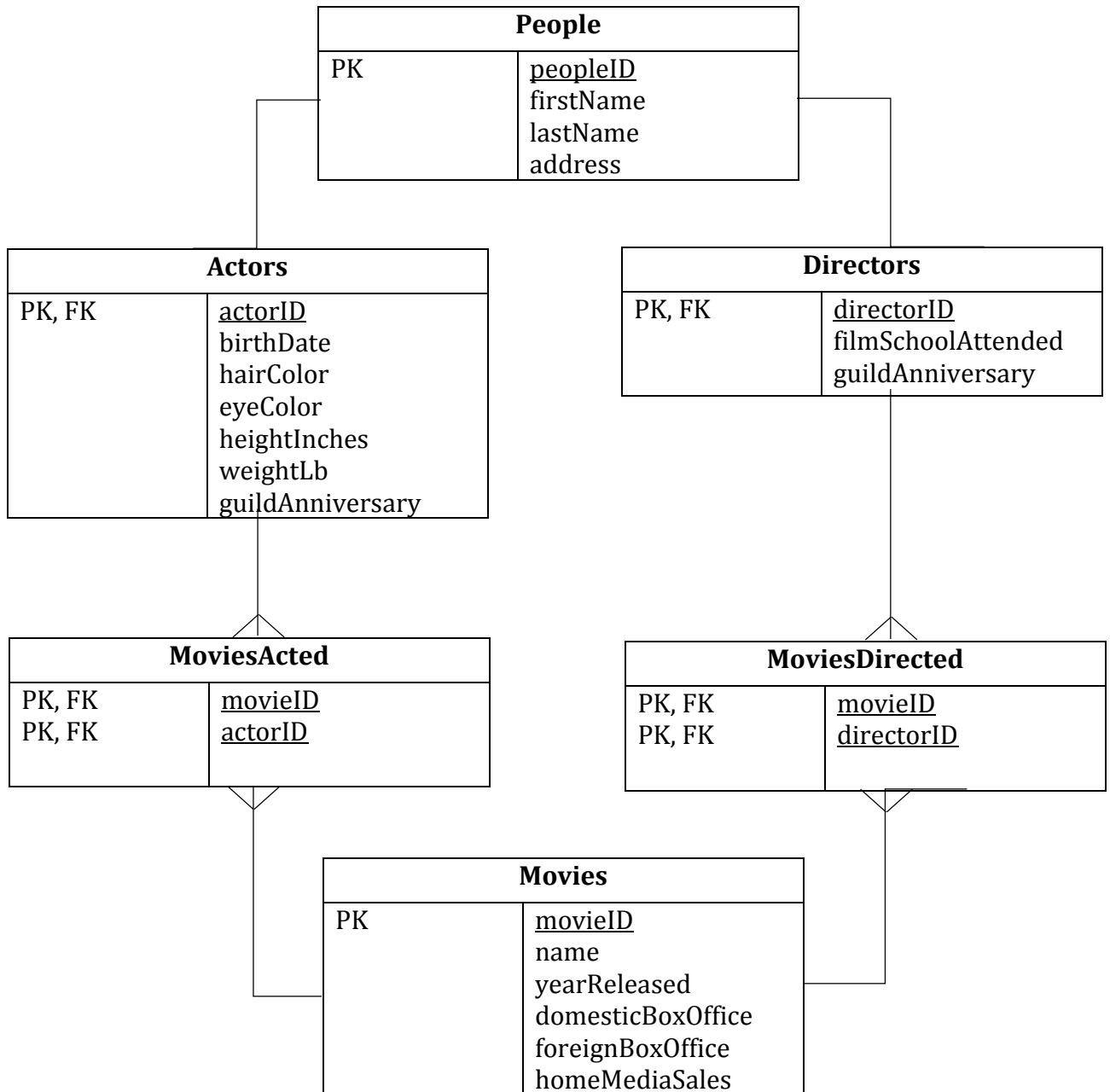| **Movies**	
PK	movieID
	name
	yearReleased
	domesticBoxOffice
	foreignBoxOffice
	homeMediaSales

2. SQL

```
drop table if exists People;
create table if not exists People (
      peopleID serial,
      firstName text not null,
      lastName text not null,
      address text not null,
  primary key (peopleID)
);

drop table if exists Actors;
create table if not exists Actors (
      actorID int not null references People (peopleID),
      birthDate date not null,
      hairColor text not null,
      eyeColor text not null,
      heightInches decimal not null,
      weightLb decimal not null,
      guildAnniversary date,
  primary key (actorID)
);

drop table if exists Directors;
create table if not exists Directors (
      directorID int not null references People (peopleID),
      filmSchoolAttended text,
      guildAnniversary date,
  primary key (directorID)
);

drop table if exists Movies;
create table if not exists Movies (
      movieID serial,
      name text not null,
      yearReleased int,
      domesticBoxOffice decimal,
      foreignBoxOffice decimal,
      homeMediaSales decimal,
  primary key (movieID)
);

drop table if exists MoviesActed;
create table if not exists MoviesActed (
      movieID int not null references Movies (movieID),
      actorID int not null references Actors (actorID),
  primary key (movieID, actorID)
);

drop table if exists MoviesDirected;
create table if not exists MoviesDirected (
      movieID int not null references Movies (movieID),
      directorID int not null references Directors (directorID),
  primary key (movieID, directorID)
```

```sql
    );


-- Inserting data in People table --
insert into People (firstName, lastName, address)
    values ('Sean', 'Connery', '007 Martini Ave.');

insert into People (firstName, lastName, address)
    values ('Steve', 'Carell', '1725 Slough Ave.');

insert into People (firstName, lastName, address)
    values ('Channing', 'Tatum', '21 Jump St.');

insert into People (firstName, lastName, address)
    values ('Terence', 'Young', '3399 North Rd.');

insert into People (firstName, lastName, address)
    values ('Guy', 'Hamilton', '2112 Bond Dr.');

insert into People (firstName, lastName, address)
    values ('Peter', 'Segal', '45 Kingston Ave.');

insert into People (firstName, lastName, address)
    values ('Phil', 'Lord', '1007 Mountain Dr.');

insert into People (firstName, lastName, address)
    values ('Chris', 'Miller', '23 Cedar St.');


-- Inserting data in Actors table --
insert into Actors (actorID, birthDate, hairColor, eyeColor,
                    heightInches, weightLb, guildAnniversary)
    values (1, '1930-08-25', 'White', 'Brown', 74, 200,
            '1953-11-12');

insert into Actors (actorID, birthDate, hairColor, eyeColor,
                    heightInches, weightLb, guildAnniversary)
    values (2, '1962-08-16', 'Brown', 'Brown', 69, 163,
            '1991-04-11');

insert into Actors (actorID, birthDate, hairColor, eyeColor,
                    heightInches, weightLb, guildAnniversary)
    values (3, '1980-04-26', 'Brown', 'Green', 73, 200,
            '2003-05-30');


-- Inserting data in Directors table --
insert into Directors (directorID, filmSchoolAttended,
                       guildAnniversary)
    values (4, 'University of Cambridge', '1939-01-24');

insert into Directors (directorID, filmSchoolAttended,
                       guildAnniversary)
    values (5, 'None', '1952-06-13');
```

```sql
insert into Directors (directorID, filmSchoolAttended,
                       guildAnniversary)
    values (6, 'None', '1994-10-16');

insert into Directors (directorID, filmSchoolAttended,
                       guildAnniversary)
    values (7, 'Dartmouth College', '2002-11-03');

insert into Directors (directorID, filmSchoolAttended,
                       guildAnniversary)
    values (8, 'Dartmouth College', '2002-08-16');


-- Inserting data in Movies table --
insert into Movies (name, yearReleased, domesticBoxOffice,
                       foreignBoxOffice, homeMediaSales)
    values ('Dr. No', 1962, 16067035.00, 43500000.00,
            23200000.00);

insert into Movies (name, yearReleased, domesticBoxOffice,
                       foreignBoxOffice, homeMediaSales)
    values ('From Russia with Love', 1963, 24800000.00,
            54100000.00, 70900000.00);

insert into Movies (name, yearReleased, domesticBoxOffice,
                       foreignBoxOffice, homeMediaSales)
    values ('Goldfinger', 1964, 51100000.00, 73800000.00,
            49500000.00);

insert into Movies (name, yearReleased, domesticBoxOffice,
                       foreignBoxOffice, homeMediaSales)
    values ('Get Smart', 2008, 130319208.00, 98423705.00,
            35597138.00);

insert into Movies (name, yearReleased, domesticBoxOffice,
                       foreignBoxOffice, homeMediaSales)
    values ('21 Jump Street', 2012, 138447667.00, 63490596.00,
            32329625.00);


-- Inserting data in MoviesActed table --
insert into MoviesActed (movieID, actorID)
    values (1, 1);

insert into MoviesActed (movieID, actorID)
    values (2, 1);

insert into MoviesActed (movieID, actorID)
    values (3, 1);

insert into MoviesActed (movieID, actorID)
    values (4, 2);
```

```
insert into MoviesActed (movieID, actorID)
    values (5, 3);


-- Inserting data in MoviesDirected table --
insert into MoviesDirected (movieID, directorID)
    values (1, 4);

insert into MoviesDirected (movieID, directorID)
    values (2, 4);

insert into MoviesDirected (movieID, directorID)
    values (3, 5);

insert into MoviesDirected (movieID, directorID)
    values (4, 6);

insert into MoviesDirected (movieID, directorID)
    values (5, 7);

insert into MoviesDirected (movieID, directorID)
    values (5, 8);


-- Query: Directors who worked actor Sean Connery --
select distinct p.firstName, p.lastName
from People p, Directors d, Movies m, MoviesDirected md
where d.directorID = md.directorID
  and m.movieID = md.movieID
  and d.directorID = p.peopleID
  and m.movieID in
        (select m.movieID
         from People p, Actors a, Movies m, MoviesActed ma
         where p.peopleID = a.actorID
           and a.actorID = ma.actorID
           and m.movieID = ma.movieID
           and p.firstName = 'Sean'
           and p.lastName = 'Connery'
        );
```

3. Functional dependencies:
   - People table
     - peopleID → firstName, lastName, address
   - Actors table
     - actorID → birthDate, hairColor, eyeColor, heightInches, weightLb, guildAnniversary
   - Directors table
     - directorID → filmSchoolAttended, guildAnniversary
   - Movies table
     - movieID → name, yearReleased, domesticBoxOffice, foreignBoxOffice, homeMediaSales