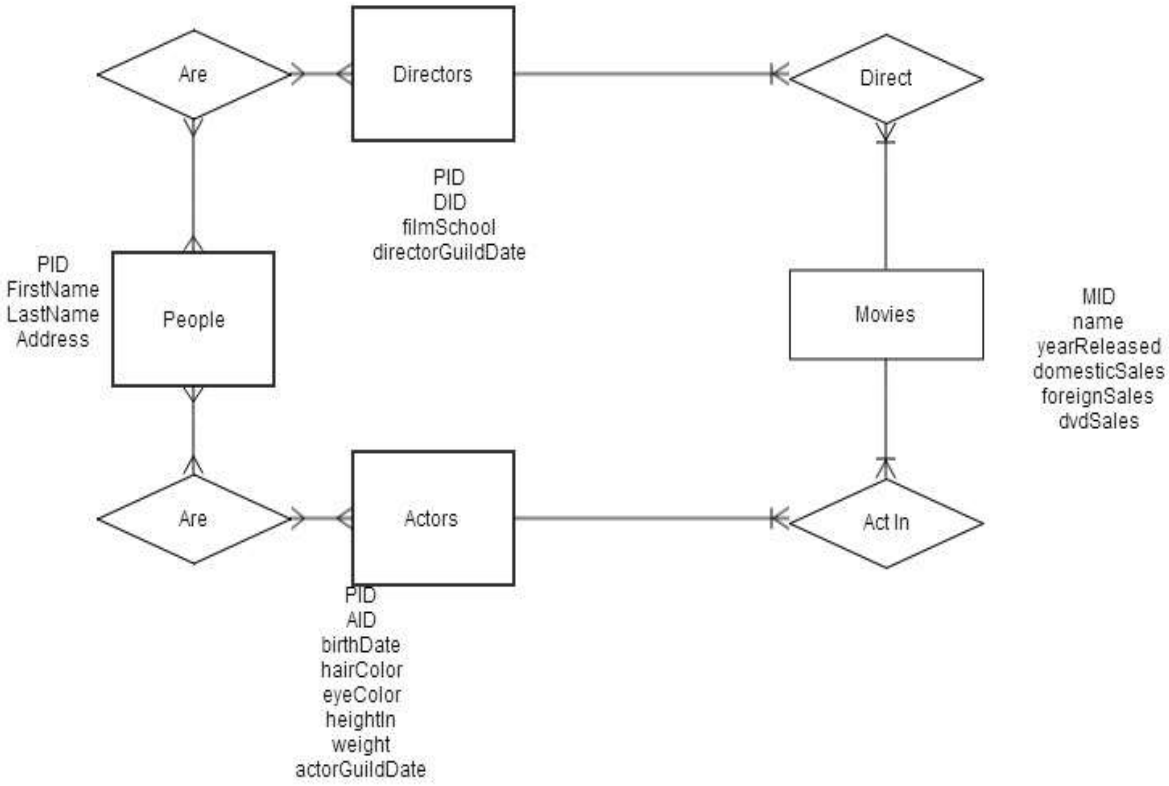
Brian Canoni

4/4/14

Normalization hw 2



2. SQL STATEMENTS

```
Drop table if exists People;
Create table if not exists people (
PeopleID integer not null,
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,
 heightIn decimal not null,
 weight decimal not null,
 actorGuildDate date,
 primary key (actorID)
);

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

drop table if exists Movies;
create table if not exists Movies (
 movieID int not null,
 name text not null,
 yearReleased int,
 domesticSales decimal,
 foreignSales decimal,
 dvdSales decimal,
 primary key (movieID)
);

Drop table if exists ActedMovies;
Create table if not exists ActedMovies(
movieID int not null references Movies (movieID),
 actorID int not null references Actors (actorID),
 primary key (movieID, actorID)
);

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

3. SQL insert statements
#inserting all the people

```
insert into People (firstName, lastName, address)
 values ('Sean', 'Connery', 'Sean Connery St');

insert into People (firstName, lastName, address)
 values ('Generic','Actor/director','Somewhere St,');

insert into People (firstName, lastName, address)
 values ('Roger','Moore','Moore St.');

insert into People (firstName, lastName, address)
 values ('Director','Dude','Director St.');

insert into People (firstName, lastName, address)
 values ('Other','Director','Director St.');

#putting people who are actors in actors
insert into Actors (actorID, birthDate, hairColor, eyeColor,  heightIn, weight,actorGuildDate)
 values (1, '1930/08/25', 'White', 'Brown', 70, 200,
 '1960');

insert into Actors (actorID, birthDate, hairColor, eyeColor,  heightIn, weight, actorGuildDate)
 values (2, '1950/01/01, 'Brown', 'Brown', 75, 190,
 '1975');


insert into Actors (actorID, birthDate, hairColor, eyeColor,  heightIn, weight, actorGuildDate)
 values (3, '1927/10/14', 'Brown, 'Brown', 70, 200,
 '1980');

## directors

insert into Directors (directorID, filmSchool,  directorGuildDate)
 values (2, 'Director School, '1940/01/01');

insert into Directors (directorID, filmSchool,  directorGuildDate)
 values (4, 'Director School2, '1941/01/01');

insert into Directors (directorID, filmSchool,  directorGuildDate)
 values (5, 'Director School3, '1942/01/01');

##movies
insert into Movies (name, yearReleased, domesticSales,  foreignSales, dvdSales)
 values ('Dr. No', 1962, 99999.99, 99998.98,
 77788777.91);
```

```
insert into Movies (name, yearReleased, domesticSales,  foreignSales, dvdSales)
 values ('movie2', 1965, 99999.99, 0098.98,
 73448777.91);


insert into Movies (name, yearReleased, domesticSales,  foreignSales, dvdSales)
 values ('movie3', 1970, 454599.99, 330098.98,
2345.91);


insert into Movies (name, yearReleased, domesticSales,  foreignSales, dvdSales)
 values ('movie4', 1975, 99555.99, 938.98,
 72227.91);


insert into Movies (name, yearReleased, domesticSales,  foreignSales, dvdSales)
 values ('movie5', 1980, 22999.99, 96698.98,
 73434777.91);


insert into ActedMovies (movieID, actorID)
 values (1, 1);
insert into ActedMovies (movieID, actorID)
 values (2, 1);
insert into ActedMovies (movieID, actorID)
 values (1,4);
insert into ActedMovies (movieID, actorID)
 values (3, 3);
insert into ActedMovies (movieID, actorID)
 values (2, 2);
insert into ActedMovies (movieID, actorID)
 values (3, 1);
insert into ActedMovies (movieID, actorID)
 values (4, 3);
insert into ActedMovies (movieID, actorID)
 values (5, 1);

insert into DirectedMovies (movieID, directorID)
values (1, 3);
 insert into DirectedMovies (movieID, directorID)
values (2, 4);
insert into DirectedMovies (movieID, directorID)
values (3, 5);


insert into DirectedMovies (movieID, directorID)
values (4, 3);
```

insert into DirectedMovies (movieID, directorID)
values (5, 4);

4. Functional dependancies within my database are
People table : peopleID is dependant on first name last name and address
Actors table: actorID is dependant on ,birthDate, hairColor, eyeColor, heightIn, weight, actorGuildDate
Directors table : director ID is depenadant on filmSchool, directorGuildDate
Movies table : MovieID is dependant on name, yearReleased, domesticSales, foreignSales, dvdSales

5. query to return all the directors Sean Connery has worked with
This should work

```
Select distinct p.firstName, p.lastName
From people p, directors d, movies m, directedMovies dm
Where d.directorID = dm.directorID
And d.directorID = p.peopleID
And m.movieID in
( ## movies sean has acted in
Select m.movieID
From people p, actors a, movies m, actedMovies am
Where p.peopleID = a.actorID
And a.actorID = am.ActorID
And m.movieID = am.movieID
And p.firstName = 'Sean'
And p.lastName = 'Connery'
);
```