Jonathan Pistilli
Normalization HW #2


1.



**actors**

| PK | **actorID** |
|---|---|
| | name |
| | address |
| | birthDate |
| | hairColor |
| | eyeColor |
| | heightInInches |
| | weight |
| | sagAnnDate |

**movies**

| PK | **movieID** |
|---|---|
| | name |
| | yearReleased |
| | domSales |
| | forSales |
| | dvdbluraySales |

**directors**

| PK | **dirID** |
|---|---|
| | name |
| | address |
| | filmSchool |
| | dguildAnnDate |

**actMov**

| PK, FK1 | **actorID** |
|---|---|
| PK, FK2 | **movieID** |

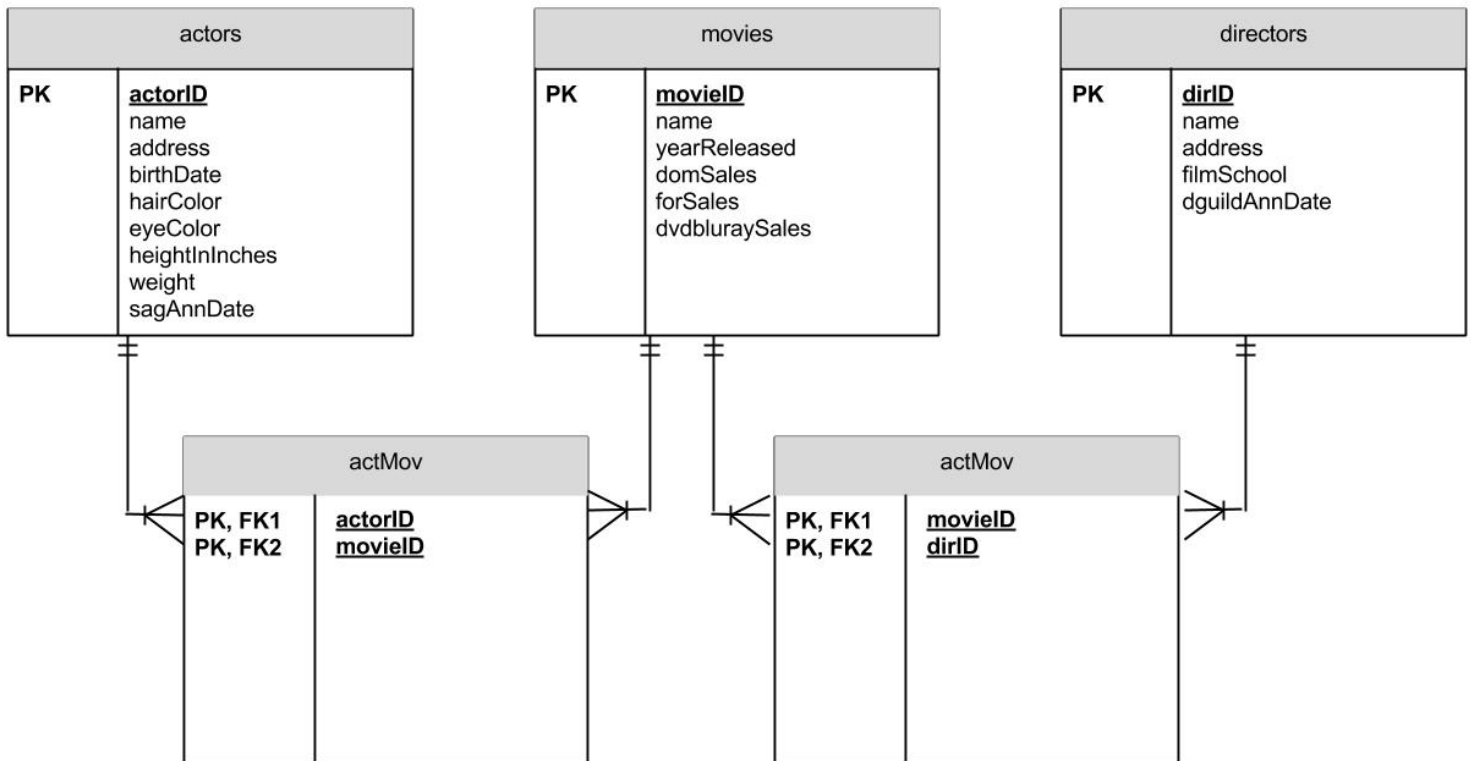**actMov**

| PK, FK1 | **movieID** |
|---|---|
| PK, FK2 | **dirID** |

2.
Create Table actors (
        actorID char (3) not null,
        name varchar (30) not null,
        address varchar (60),
        birthDate date,
        hairColor varchar(15),
        eyeColor varchar(15),
        heightInInches int,
        weight int,
        sagAnnDate date,
primary key (actorID)
);

```
Create Table movies (
        movieID char(3) not null,
        name varchar (30) not null,
        yearReleased int,
        domSales int,
        forSales int,
        dvdbluraySales int,
primary key (movieID)
);

create table directors (
        dirID char(3) not null,
        name varchar (30) not null,
        address varchar (60),
        filmSchool varchar (30),
        dguildAnnDate date,
primary key (dirID)
);

create table actMov (
        actorID char (3) references actors(actorID),
        movieID char (3) references movies(movieID)
);

create table movDir (
        movieID char (3) references movies(movieID),
        dirID char (3) references directors(dirID)
);
```

3.

```
insert into actors ( actorID, name, address, birthDate, hairColor, eyeColor,
heightInInches, weight, sagAnnDate )
        values ( 'a01', 'Sean Connery', '375 Actor Street', '08/25/1930', 'grey',
        'brown', 74, 180, '1/12/1975' );

insert into actors ( actorID, name, address, birthDate, hairColor, eyeColor,
heightInInches, weight, sagAnnDate )
        values ( 'a02', 'Pierce Brosnan', '12 Bungie Boulevard', '09/24/1950',
        'brown', 'blue', 75, 200, '1/17/1995' );

insert into actors ( actorID, name, address, birthDate, hairColor, eyeColor,
heightInInches, weight, sagAnnDate )
        values ( 'a03', 'Tom Hanks', '45 Wilson Road', '12/14/1960', 'brown', 'green',
        79, 220, '1/23/1993' );
```

```sql
insert into actors ( actorID, name, address, birthDate, hairColor, eyeColor,
heightInInches, weight, sagAnnDate )
        values ( 'a04', 'George Clooney', '24 Robin Lane', '05/06/1961', 'peppered',
        'brown', 72, 200, '1/15/1999' );

insert into movies (movieID, name, yearReleased, domSales, forSales,
dvdbluraySales)
        values ( 'm01', '007', 1975, 195432, 123984, 2138723 );

insert into movies (movieID, name, yearReleased, domSales, forSales,
dvdbluraySales)
        values ( 'm02', '007: Golden Eye', 1995, 2309432, 2345613, 64234126 );

insert into movies (movieID, name, yearReleased, domSales, forSales,
dvdbluraySales)
        values ( 'm03', 'Castaway', 2000, 195432, 123984, 2138723 );

insert into movies (movieID, name, yearReleased, domSales, forSales,
dvdbluraySales)
        values ( 'm04', 'Batman and Robin', 1999, 195432, 123984, 2138723 );

insert into directors ( dirID, name, address, filmSchool, dguildAnnDate )
        values ( 'd01', 'Terance Young', '375 Actor Street', 'NYU', '1/25/85' );

insert into directors ( dirID, name, address, filmSchool, dguildAnnDate )
        values ( 'd02', 'Martin Campbell', '13 Golden Eye Road', 'Prestige', '1/23/95');

insert into directors ( dirID, name, address, filmSchool, dguildAnnDate )
        values ( 'd03', 'Robert Zemeckis', '907 Chuck Avenue', 'School', '1/24/2000' );

insert into directors ( dirID, name, address, filmSchool, dguildAnnDate )
        values ( 'd04', 'Joel Schumacher', '22 Ivy Lane', 'Iflakis', '1/29/1999' );

insert into actMov ( actorID, movieID)
        values ( 'a01', 'm01');

insert into actMov ( actorID, movieID)
        values ( 'a02', 'm02');

insert into actMov ( actorID, movieID)
        values ( 'a03', 'm03');

insert into actMov ( actorID, movieID)
        values ( 'a04', 'm04');
```

insert into movDir ( movieID, dirID )
        values ( 'm01', 'd01');

insert into movDir ( movieID, dirID )
        values ( 'm02', 'd02');

insert into movDir ( movieID, dirID )
        values ( 'm03', 'd03');

insert into movDir ( movieID, dirID )
        values ( 'm04', 'd04');


4. Functional Dependencies

actorID → name, address, birthdate, hairColor, eyeColor, heighInInches, weight,
        sagAnnDate

movieID → name, yearReleased, domSales, forSales, dvdbluraySales

dirID → name, address, filmSchool, dguildAnnDate

For the tables actMov and movDir, there are no functional dependencies because the
foreign keys in each of the tables create composite primary keys, with no data
within the table relying on the primary key.

5. Selecting the directors that have worked with Sean Connery

```
select d.name
from actors a, directors d, movies m, actMov am, movDir md
where a.name = 'Sean Connery'
  AND a.actorID = am.actorID
  AND am.movieID = m.movieID
  AND m.movieID = md.movieID
  AND md.dirID = d.dirID;
```