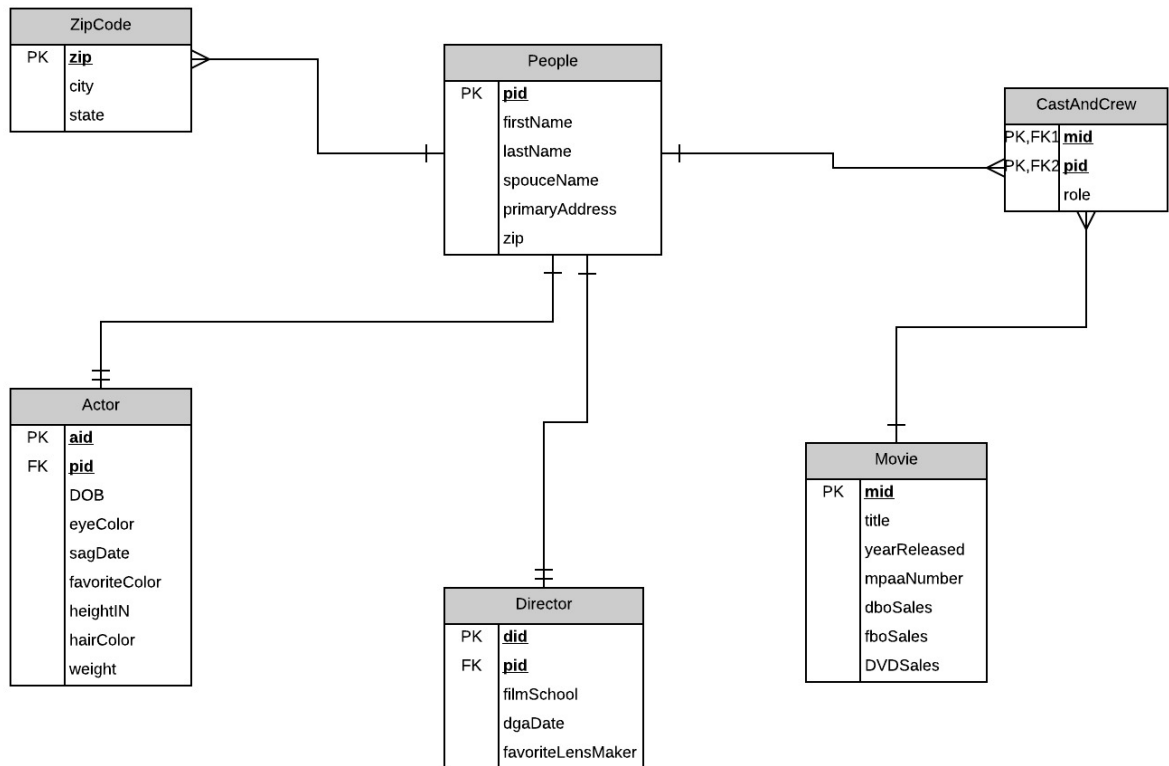


Blaise Spinelli  
Alan Labouseur  
Lab 8: Normalization Two  
28 March 2017

**1. My fully decorated and aesthetically beautiful E/R Diagram:**



## **2. SQL Create Statements**

```
CREATE TABLE People (  
    pid char(4) NOT NULL,  
    firstName text NOT NULL,  
    lastName text NOT NULL,  
    spouseName text NOT NULL,  
    primaryAddress text NOT NULL,  
    zip int NOT NULL,  
    primary key (pid)  
);
```

```
Create Table Actor (  
    aid char(4) NOT NULL,  
    pid char(4) NOT NULL REFERENCES People (pid),  
    DOB date NOT NULL,  
    eyeColor text NOT NULL,  
    sagDate date,  
    favoriteColor text,  
    heightIN integer NOT NULL,  
    hairColor text,  
    weight integer NOT NULL,  
    primary key (aid)  
);
```

```
CREATE TABLE Director (  
    did char(4) NOT NULL,  
    pid char(4) NOT NULL REFERENCES People (pid),  
    filmSchool text,  
    dgaDate date,  
    favoriteLensMaker text,  
    primary key (did)  
);
```

```
CREATE TABLE Movie (  
    mid char(4) NOT NULL,  
    title text NOT NULL,  
    yearReleased date NOT NULL,  
    mpaaNumber integer NOT NULL,  
    dboSales integer,  
    fboSales integer,  
    DVDSales integer,  
    primary key (mid)  
);
```

```
CREATE TABLE CastAndCrew (
  mid char(4) NOT NULL REFERENCES Movie (mid),
  pid char(4) NOT NULL REFERENCES People (pid),
  role text NOT NULL,
  primary key (mid,pid)
);
```

```
CREATE TABLE Zipcode (
  zip char(5) NOT NULL,
  city text NOT NULL,
  state text NOT NULL,
  primary key (zip)
);
```

### ■ Drop Table Statement

```
DROP TABLE IF EXISTS People CASCADE;
DROP TABLE IF EXISTS Zipcode CASCADE;
DROP TABLE IF EXISTS Actor CASCADE;
DROP TABLE IF EXISTS Director CASCADE;
DROP TABLE IF EXISTS Movie CASCADE;
DROP TABLE IF EXISTS CastandCrew CASCADE;
```

### 3. Functional Dependencies (Table names are in bold)

#### ❖ **People**

pid → firstName, lastName, spouseName, primaryAddress, zip

#### ❖ **Actor**

pid → aid

aid → DOB, eyeColor, sagDate, favoriteColor, heightIN, hairColor, weight

#### ❖ **Director**

pid → did

did → filmSchool, dgaDate, favoriteLensMaker

#### ❖ **Movies**

mid → title, yearReleased, mpaaNumber, dboSales, fboSales, DVDSales

#### ❖ **Zipcode**

zip → city, state

#### ❖ **CastandCrew**

mid, pid → role

#### 4. Insert Into & Query Statement

## Insert

```
INSERT INTO People (pid,firstName,lastName,spouceName,primaryAddress,zip)
Values (0001, 'Sean', 'Connery', 'Micheline','123 cherry lane', 11220) ;
```

```
INSERT INTO ACTOR
(aid,pid,DOB,eyeColor,sagDate,favoriteColor,heightIN,hairColor,weight)
Values (0001,0001,'Jan-01-1999','blue','Jan-01-1999','red','72','brown','200')
```

Insert INTO Movie  
values (0001,'bond', 'jan-01-2000', 1, 2,3,4)

```
INSERT INTO CastandCrew
values(0001, 0001, 'actor')
```

```
INSERT INTO CastandCrew
values(0001,0002, 'director')
```

```
INSERT INTO People (pid,firstName,lastName,spouceName,primaryAddress,zip)
Values (0002, 'Guy', 'Hamilton', 'Kerima','124 cherry lane', 11221);
```

```
INSERT INTO Director (did,pid,filmSchool,dgaDate,favoriteLensMaker)
Values (0001,0002, 'Harvard', 'Jan-01-1965', 'Kodiak');
```

## Query

```
select * from People
where pid in (select pid
              from Director
              where did in (select did
                           from CastandCrew
                           where mid in (select mid
                                         from Movie
                                         where mid in(select mid
                                                         from CastandCrew
                                                         where pid in (select pid
                                                                     from Actor
                                                                     where pid in (select pid
                                                                                     from People
                                                                                     where firstName='Sean' and
lastName='Connery'
```

); ) ) )