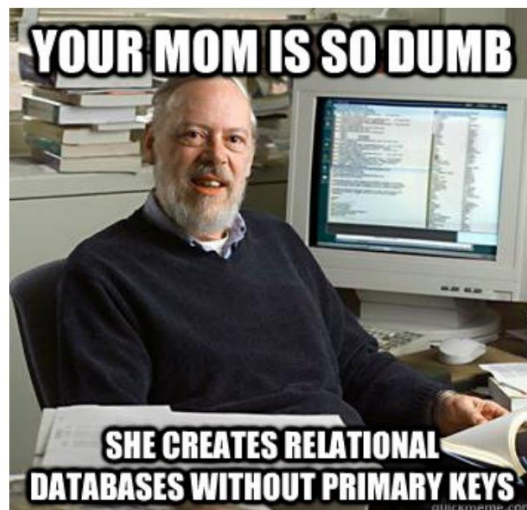
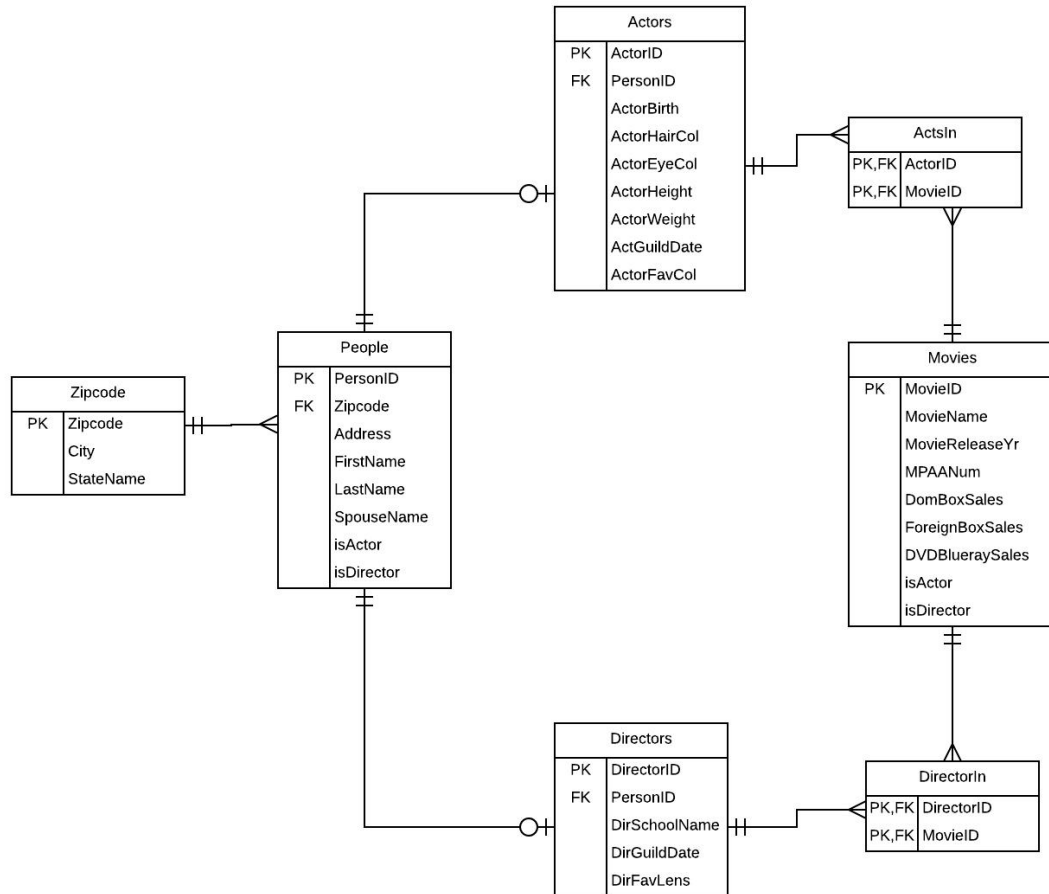


Eric Schmidt

1)



2)

-- Eric Schmidt --

DROP TABLE IF EXISTS Actors;

DROP TABLE IF EXISTS Directors;

DROP TABLE IF EXISTS Movies;

DROP TABLE IF EXISTS Productions;

DROP TABLE IF EXISTS ZipCode;

DROP TABLE IF EXISTS People;

DROP TABLE IF EXISTS ActsIn;

DROP TABLE IF EXISTS DirectsIn;

-- Zipcodes --

CREATE TABLE Zipcode (

 zipcodes char(5) not null,

 city text,

 stateName text,

 primary key(zipcodes)

);

-- People --

CREATE TABLE People(

 PersonID char(5) not null,

 zipcodes char(5) not null references ZipCodes(zipcodes),

 FirstName text,

 LastName text,

 SpouseName text,

 isActor boolean,

 isDirector boolean,

```
    primary key(PersonID)
);
```

-- Actors --

```
CREATE TABLE Actors (
    ActorID                char(3) not null,
    PersonID               char(5) not null references People(PersonID),
    ActorBirth             date,
    ActorHairColor         text,
    ActorEyeColor          text,
    ActorHeight            text,
    ActorWeight            text,
    ActorFavColor          text,
    ActorGuildDate         date,
    primary key(ActorID)
);
```

-- ActsIn --

```
CREATE TABLE ActsIn(
    ActorID char(3) not null references Actors(ActorID),
    Movies char(3) not null references Movies(MovieID),
    primary key(ActorID, MovieID)
);
```

-- Directors --

```
CREATE TABLE Directors (
    DirectorID            char(3) not null,
    PersonID              char(5) not null references People(PersonID),
```

```
DirSchoolName          text,
DirGuildDate           text,
DirFavLens             text,
primary key(DirectorID)
);
```

-- DirectsIn --

```
CREATE TABLE DirectsIn(
    DirectorID char(3) not null references Directors(DirectorID),
    Movies char(3) not null references Movies(MovieID),
    primary key(DirectorID, MovieID)
);
```

-- Movies --

```
CREATE TABLE Movies (
    MovieID          char(3) not null,
    MovieName        text,
    MovieReleaseYr    int,
    MPAAEnum         int,
    DomBoxSales       numeric(12,2),
    ForeignBoxSales   numeric(12,2),
    DVDBlueraySales   numeric(12,2),
    primary key(MovieID)
);
```

-- EonProductions --

```
CREATE TABLE EonProductions (
```

ActorsID char(5) not null references Actors(ActorID),
 DirectorID char(5) not null references Directors(DirectorID),
 MovieID char(3) not null references Movies(MovieID),
 primary key(ActorsID,DirectorID,MovieID)
);

3) Functional Dependencies

- Persons
 - PersonID → Address, FirstName, LastName, SpouseName, isActor, isDirector
 - Zipcode → PersonID
- Actors
 - PersonID → ActorID
 - ActorID → ActorBirth, ActorHairCol, ActorEyeCol, ActorHeight, ActorWeight, ActGuildDate, ActorFavCol
- Directors
 - PersonID → DirectorID
 - DirectorID → DirSchoolName, DirGuildDate, DirFavLens
- ActsIn
 - MovieID → ActorID
- DirectsIn
 - MovieID → DirectorID
- Movies
 - MovieID → MovieName, MovieReleaseYr, MPAANum, DomBoxSales, ForeignBoxSales, DVDBlueraySales, isActor, isDirector
- Zipcode
 - Zipcodes → City, Statename

4)

SELECT *

FROM PersonID

WHERE PersonID IN (

 SELECT PersonID

 FROM Directors

 WHERE DirectorID IN(

 SELECT DirectorID

 FROM DirectsIn

 WHERE MovieID IN (

 SELECT MovieID

 FROM ActsIn

 WHERE ActorID IN(

 SELECT ActorID

 FROM Actors

 WHERE PersonID IN(

 SELECT PersonID

 FirstName = 'Sean' and

 LastName = 'Connery'

)

)

)

)

)