

Chris Piccirillo

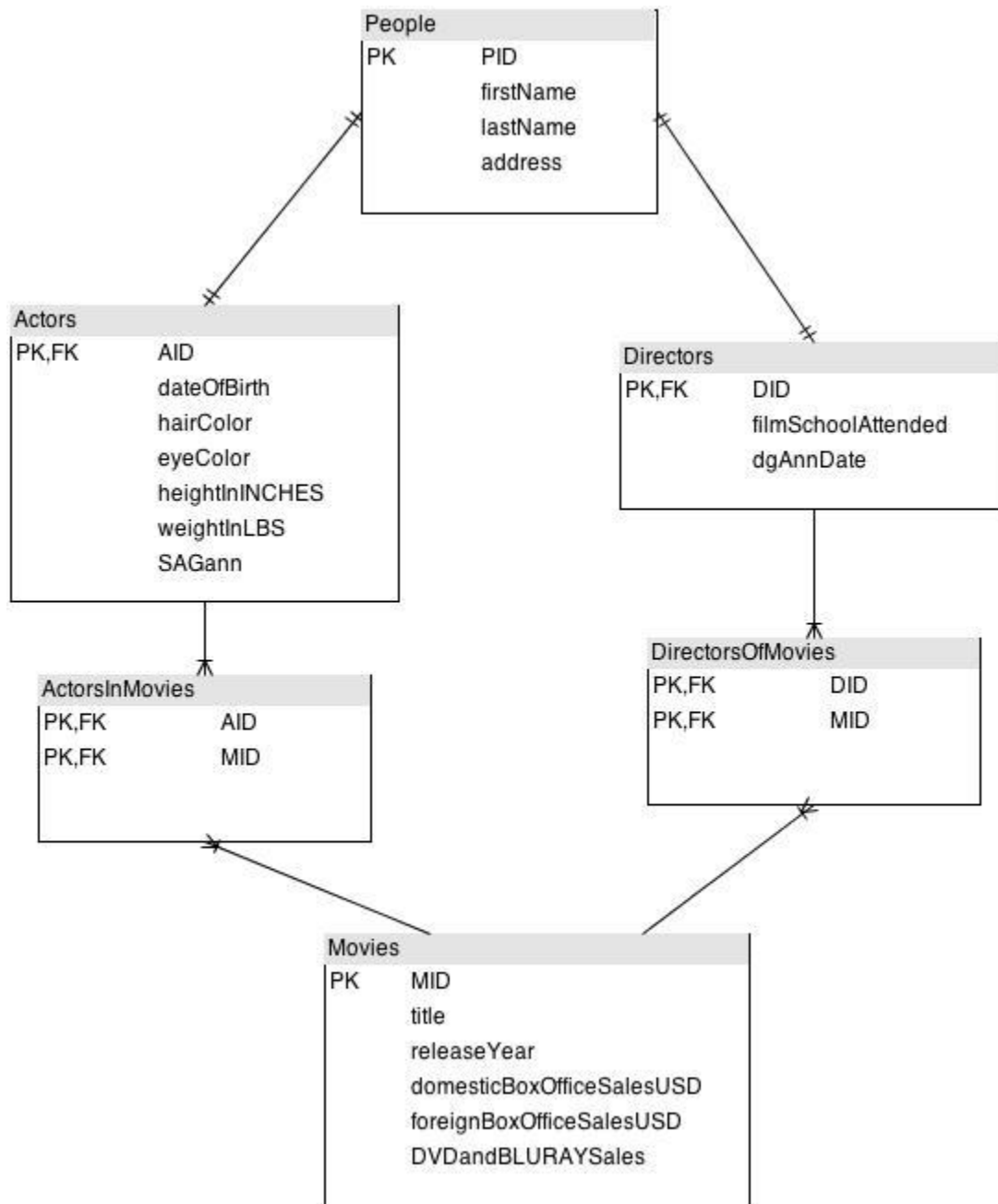
Professor Labouseur

Database Management

November 3, 2013

Normalization #2 HW

E-R Diagram



Create and Insert Statements (SQL)

-----Create People Table-----

DROP TABLE IF EXISTS People;

```
CREATE TABLE People (  
    pid char(4) NOT NULL PRIMARY KEY,  
    firstName varchar NOT NULL,  
    lastName varchar NOT NULL,  
    address varchar NOT NULL  
  
    );
```

-----Insert into People Table-----

```
INSERT INTO People( pid, firstName, lastName, address )  
VALUES('p001', 'Michael', 'Jackson', '142 Candy Road');
```

```
INSERT INTO People( pid, firstName, lastName, address )  
VALUES('p002', 'Jon', 'Snow', '515 Smithington Lane');
```

```
INSERT INTO People( pid, firstName, lastName, address )  
VALUES('p003', 'Sean', 'Connery', '615 Yolanda Avenue');
```

-----Create Actors Table-----

DROP TABLE IF EXISTS Actors;

```
CREATE TABLE Actors (  
    aid char(4) NOT NULL PRIMARY KEY,
```

```
dateofBirth date NOT NULL,  
    hairColor varchar NOT NULL,  
    eyeColor varchar NOT NULL,  
    heightInInches int NOT NULL,  
    weightInLBS int NOT NULL,  
    SAGann date NOT NULL,  
    Foreign Key(aid) References People(pid)  
  
);
```

-----Insert Into Actors Table-----

```
INSERT INTO Actors( aid, dateOfBirth, hairColor, eyeColor, heightInInches, weightInLBS, SAGann)  
VALUES('p001', '1971-02-24', 'Blonde' , 'Brown', 54, 130, '2004-05-22');
```

```
INSERT INTO Actors( aid, dateOfBirth, hairColor, eyeColor, heightInInches, weightInLBS, SAGann)  
VALUES('p002', '1970-08-28', 'Blonde' , 'Blue', 60, 150, '2002-05-23');
```

```
INSERT INTO Actors( aid, dateOfBirth, hairColor, eyeColor, heightInInches, weightInLBS, SAGann)  
VALUES('p003', '1980-02-19', 'Brunette' , 'Green', 56, 190, '2007-11-10');
```

-----Create Directors Table-----

```
DROP TABLE IF EXISTS Directors;
```

```
CREATE TABLE Directors (
```

```
did char(4) NOT NULL PRIMARY KEY,
```

```
filmSchoolAttended varchar NOT NULL,  
dgAnnDate date NOT NULL,  
FOREIGN KEY (did) references People (pid)
```

```
);
```

-----Insert Into Directors Table-----

```
INSERT INTO Directors( did, filmSchoolAttended, dgAnnDate)  
VALUES('p001', 'SmithHavens School', '2010-09-21');  
INSERT INTO Directors( did, filmSchoolAttended, dgAnnDate)  
VALUES('p002', 'MikeJones School', '2005-04-12');  
INSERT INTO Directors( did, filmSchoolAttended, dgAnnDate)  
VALUES('p003', 'RobMitolasSchool', '2004-01-20');
```

-----Create Movies Table-----

```
DROP TABLE IF EXISTS Movies;
```

```
CREATE TABLE Movies(
```

```
mid char(4) NOT NULL PRIMARY KEY,  
title varchar NOT NULL,  
releaseYear int NOT NULL,  
domesticBoxOfficeSalesUSD float NOT NULL,  
foreignBoxOfficeSalesUSD float NOT NULL,  
DVDandBLURAYSales float NOT NULL  
);
```

-----Insert Into Movies Table-----

```
INSERT INTO Movies( mid, title, releaseYear,domesticBoxOfficeSalesUSD
,foreignBoxOfficeSalesUSD,DVDandBLURAYSales )
VALUES('m001', 'Aliens In Space', '2000' ,200000.00, 500000.00, 20000.00);
```

```
INSERT INTO Movies( mid, title, releaseYear,domesticBoxOfficeSalesUSD
,foreignBoxOfficeSalesUSD,DVDandBLURAYSales )
VALUES('m002', 'Giant HotDogs', '1995' ,300000.00, 100000.00, 10000.00);
```

```
INSERT INTO Movies( mid, title, releaseYear,domesticBoxOfficeSalesUSD
,foreignBoxOfficeSalesUSD,DVDandBLURAYSales )
VALUES('m003', 'Miles Adventure', '2003' ,400000.00, 200000.00, 5000.00);
```

-----Create ActorsInMovies Table-----

```
DROP TABLE IF EXISTS ActorsInMovies;
```

```
CREATE TABLE ActorsInMovies(
aid char(4) NOT NULL,
mid char(4) NOT NULL,
primary key(aid,mid),
foreign key(aid) references Actors(aid),
foreign key(mid) references Movies(mid)
);
```

-----Insert Into ActorsInMovies Table-----

```
INSERT INTO ActorsInMovies(aid,mid)
VALUES('p001','m001');
INSERT INTO ActorsInMovies(aid,mid)
```

```
VALUES('p002','m002');
```

```
INSERT INTO ActorsInMovies(aid,mid)
```

```
VALUES('p003','m003');
```

```
-----Create DirectorsOfMovies Table-----
```

```
DROP TABLE IF EXISTS DirectorsOfMovies;
```

```
CREATE TABLE DirectorsOfMovies(
```

```
did char(4) NOT NULL,
```

```
mid char(4) NOT NULL,
```

```
primary key(did,mid),
```

```
foreign key(did) references Directors(did),
```

```
foreign key(mid) references Movies(mid)
```

```
);
```

```
-----Insert Into DirectorsInMovies Table-----
```

```
INSERT INTO DirectorsOfMovies(did,mid)
```

```
VALUES('p001','m001');
```

```
INSERT INTO DirectorsOfMovies(did,mid)
```

```
VALUES('p002','m002');
```

```
INSERT INTO DirectorsOfMovies(did,mid)
```

```
VALUES('p003','m003');
```

Functional Dependencies For Each Table

People Table

pid → firstname, lastname, address

Actors Table

aid → birthDate, hairColor, eyeColor, heightinINCHES, weightinLBS, SAGann

Directors Table

did → filmSchoolAttended, directorsGuildAnniversaryDate

Movies Table

Movies → name, yearReleased, domesticBoxOfficeSalesUSD, foreignBoxOfficeSalesUSD, DVDandBluRaySalesUSD

ActorsInMovies and **DirectorsOfMovies** don't have dependencies, just composite keys

SQL Query to return all the directors with whom actor "Sean Connery" has worked

```
SELECT p.firstname, p.lastname
FROM people p
WHERE p.pid IN
    (SELECT d.did
     FROM directorsOfMovies d
     WHERE mid IN
        (SELECT m.mid
         FROM movies m
         WHERE mid IN
            (SELECT mid
             FROM actorsInMovies
             WHERE aid IN
                (SELECT p.pid
                 FROM people p
                 WHERE firstname = 'Sean' AND lastname = 'Connery'))));
```