Robert Mitola

Professor Labouseur

Database Management

4 November 2013

Normalization Homework 2

People

PK	pid
	fname
	lname
	address

Actors

PK, FK	aid
	birthday
	hairColor
	eyeColor
	heightIN
	weightLBS
	SAGanniversary

Directors

PK, FK	did
	filmSchool
	DGanniversary

Actors in Movies

PK, FK	aid
PK, FK	mid

Directors of Movies

PK, FK	did
PK, FK	mid

Movies

PK	mid
	name
	yearReleased
	DBOsalesUSD
	FBOsalesUSD
	diskSalesUSD
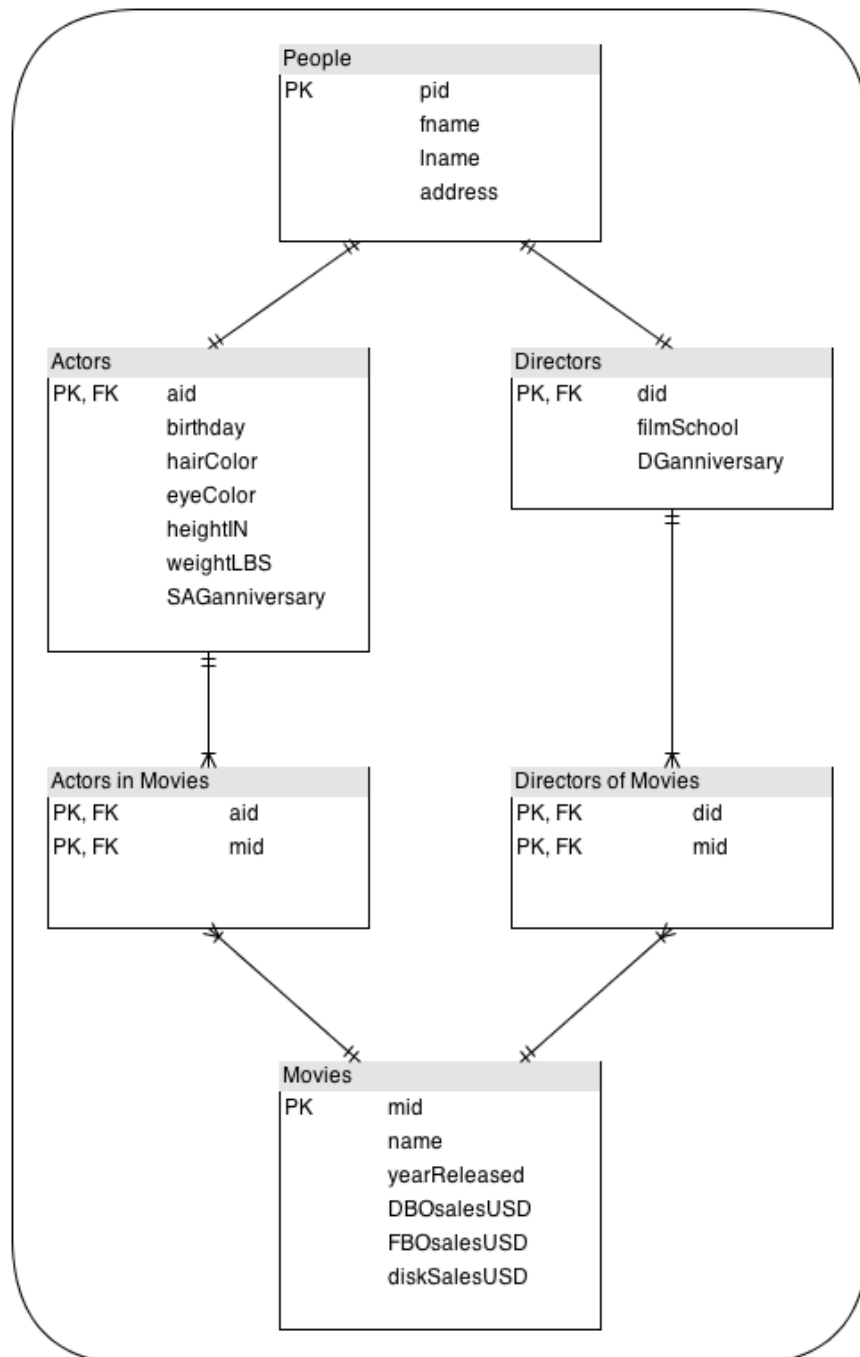
```
-- Creating the Tables --
-- People --
DROP TABLE IF EXISTS People;
CREATE TABLE People(
        pid char(4) not null,
        fname varchar(225) not null,
        lname varchar(225) not null,
        address varchar(225) not null,
 primary key(pid)
        );

-- Actors --
DROP TABLE IF EXISTS Actors;
CREATE TABLE Actors(
        aid char(4) not null,
        birthday date not null,
        hairColor varchar(225),
        eyeColor varchar(225),
        heightIN int not null,
        weightLBS int not null,
        SAGanniversary date,
primary key(aid),
foreign key(aid) references People(pid)
        );

-- Directors --
DROP TABLE IF EXISTS Directors;
CREATE TABLE Directors(
        did char(4) not null,
        filmSchool varchar(225),
        DGanniversary date,
primary key(did),
foreign key(did) references People(pid)
        );

--Movies--
DROP TABLE IF EXISTS Movies;
CREATE TABLE Movies(
        mid char(4) not null,
        name varchar(225) not null,
        yearReleased int not null,
        DBOsalesUSD float,
        FBOsalesUSD float,
        diskSalesUSD float,
```

```
primary key(mid)
        );

-- Actors in Movies--
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)
        );

-- Directors of Movies --
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)
        );


-- Inserting Values into the Tables --
-- People --
INSERT INTO People(pid, fname, lname, address)
        VALUES('p001', 'Robert', 'Mitola', '3399 North Road');
INSERT INTO People(pid, fname, lname, address)
        VALUES('p002', 'Samantha', 'Smith', '3399 North Road');
INSERT INTO People(pid, fname, lname, address)
        VALUES('p003', 'Jon', 'Mason', '123 ABC Lane');
INSERT INTO People(pid, fname, lname, address)
        VALUES('p004', 'Benjamin', 'Maxwell', '777 Church Street');

-- Actors --
INSERT INTO Actors(aid, birthday, hairColor, eyeColor, heightIN, weightLBS, SAGanniversary)
        VALUES('p001', '1994-03-11', 'brown', 'brown', 71, 150, '2000-04-22');
INSERT INTO Actors(aid, birthday, hairColor, eyeColor, heightIN, weightLBS, SAGanniversary)
        VALUES('p002', '1980-04-13', 'blue', 'blonde', 64, 130, '1999-05-12');
INSERT INTO Actors(aid, birthday, hairColor, eyeColor, heightIN, weightLBS, SAGanniversary)
        VALUES('p003', '1900-11-11', 'green', 'red', 78, 175, '1930-10-23');

-- Directors --
```

```sql
INSERT INTO Directors(did, filmSchool, DGanniversary)
        VALUES('p002', 'UCLA', '1999-01-02');
INSERT INTO Directors(did, filmSchool, DGanniversary)
        VALUES('p003', 'NYU', '1960-12-22');
INSERT INTO Directors(did, filmSchool, DGanniversary)
        VALUES('p004', 'Columbia University', '1975-05-06');

-- Movies --
INSERT INTO Movies(mid, name, yearReleased, DBOsalesUSD, FBOsalesUSD, diskSalesUSD)
        VALUES('m001', 'The Best Movie Ever', 2005, 9000000.00, 999999.99, 987678465.87);
INSERT INTO Movies(mid, name, yearReleased, DBOsalesUSD, FBOsalesUSD, diskSalesUSD)
        VALUES('m002', 'The Worst Movie Ever', 2006, 0.01, 0.01, 0.99);
INSERT INTO Movies(mid, name, yearReleased, DBOsalesUSD, FBOsalesUSD, diskSalesUSD)
        VALUES('m003', 'Once Upon a Time', 1940, 700000.00, 200000.50, 300000.75);
INSERT INTO Movies(mid, name, yearReleased, DBOsalesUSD, FBOsalesUSD, diskSalesUSD)
        VALUES('m004', 'Forgotten Galaxy', 1990, 550000.00, 90078.99, 400550.20);

-- Actors in Movies --
INSERT INTO ActorsInMovies(aid, mid)
        VALUES('p001', 'm001');
INSERT INTO ActorsInMovies(aid, mid)
        VALUES('p002', 'm001');
INSERT INTO ActorsInMovies(aid, mid)
        VALUES('p001', 'm002');
INSERT INTO ActorsInMovies(aid, mid)
        VALUES('p003', 'm003');
INSERT INTO ActorsInMovies(aid, mid)
        VALUES('p003', 'm004');

-- Directors of Movies--
INSERT INTO DirectorsOfMovies(did, mid)
        VALUES('p002', 'm001');
INSERT INTO DirectorsOfMovies(did, mid)
        VALUES('p002', 'm002');
INSERT INTO DirectorsOfMovies(did, mid)
        VALUES('p003', 'm003');
INSERT INTO DirectorsOfMovies(did, mid)
        VALUES('p004', 'm004');
INSERT INTO DirectorsOfMovies(did, mid)
        VALUES('p004', 'm001');
```

Functional Dependencies for Each Table

People
Pid → fname, lname, address

Actors
Aid→ birthday, hairColor, eyeColor, heightIN, weightLBS, SAGanniversary

Directors
did → filmSchool, DGanniversary

Movies
Mid → name, yearReleased, DBOsalesUSD, FBOsalesUSD, diskSalesUSD

Tables ActorsInMovies and DirectorsOfMovies do not have dependencies. They only have composite keys.

```
--Sean Connery Search Statement--
SELECT p.fname, p.lname
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 fname = 'Sean'
                                AND lname = 'Connery'
                                )
                        )
                )
        );
```