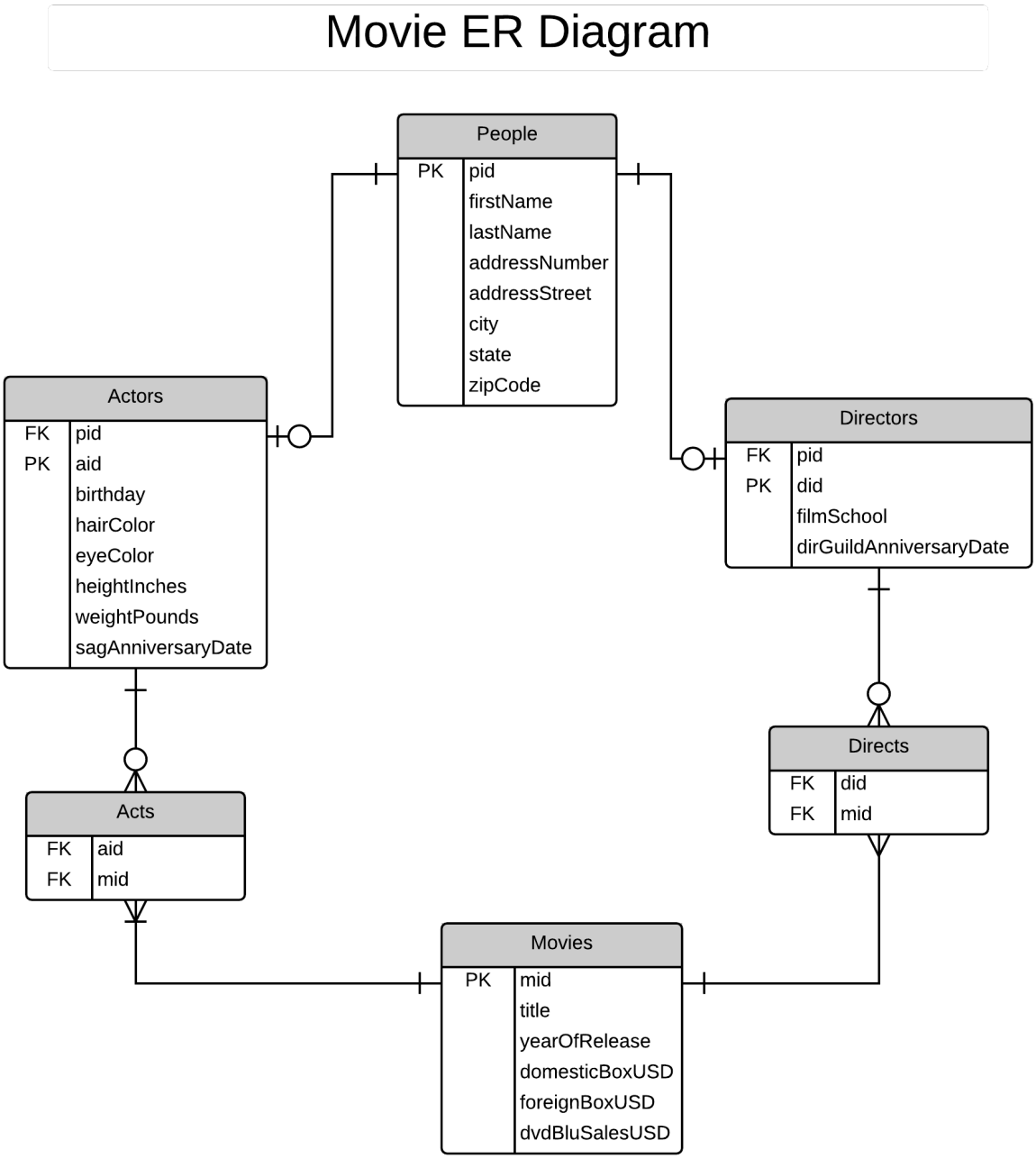
Nicholas Russell
Professor Labouseur
4 November 2014
Database Systems Management

Lab 8: Normalization Two

1) Here is an ER Diagram of the Data:



PEOPLE DATA

Here are the SQL Statements for the first table, People:

```
CREATE TABLE people(
        pid char(3) NOT NULL,
        firstName text,
        lastName text,
        addressNumber int,
        addressStreet text,
        city text,
        state text,
        zipCode int,
        primary key(pid)
        );


INSERT INTO people(pid, firstName, lastName, addressNumber, addressStreet, city, state,
        zipCode)
VALUES ('p01', 'Sean', 'Connery', 6220, 'Del Valle Drive', 'Los Angeles', 'CA', 90048),
        ('p02', 'Joseph', 'Gordon-Levitt', 8820, 'Wilshire Blvd', 'Beverly Hills', 'CA', 90211),
        ('p03', 'Joaquin', 'Phoenix', 9171, 'Wilshire Blvd', 'Beverly Hills', 'CA', 90210),
        ('p04', 'Christopher', 'Nolan', 10880, 'Wilshire Blvd', 'Beverly Hills', 'CA', 90024),
        ('p05', 'Leonardo', 'Dicaprio', 9225, 'Sunset Blvd', 'West Hollywood', 'CA', 90069),
        ('p06', 'Mark', 'Ruffalo', 9150, 'Main Street', 'Tucson', 'AZ', 85701),
        ('p07', 'Bennett', 'Miller', 1250, 'Park Avenue', 'New York', 'NY', 10021),
        ('p08', 'Javier', 'Bardem', 12, 'Herring Street', 'Harrington Park', 'NJ', 07640),
        ('p09', 'Terence', 'Young', 11, 'Broadway Avenue', 'Port Richey', 'FL', 34668),
        ('p10', 'Joel', 'Coen', 30, 'East 10th Street', 'New York', 'NY', 10003),
        ('p11', 'Ethan', 'Coen', 30, 'East 10th Street', 'New York', 'NY', 10003),
        ('p12', 'Spike', 'Jonze', 20, 'Parva Ave', 'Los Angeles', 'CA', 90027);
```

The following statements produce a table that looks like the following:

```
select *
from people;
```

	pid character(3)	firstname text	lastname text	addressnumber integer	addressstreet text	city text	state text	zipcode integer
1	p01	Sean	Connery	6220	Del Valle D	Los	CA	90048
2	p02	Joseph	Gordon-l	8820	Wilshire Bl	Beve	CA	90211
3	p03	Joaquin	Phoenix	9171	Wilshire Bl	Beve	CA	90210
4	p04	Christop	Nolan	10880	Wilshire Bl	Beve	CA	90024
5	p05	Leonardo	Dicapric	9225	Sunset Blvd	Wes	CA	90069
6	p06	Mark	Ruffalo	9150	Main Street	Tuc	AZ	85701
7	p07	Bennett	Miller	1250	Park Avenue	New	NY	10021
8	p08	Javier	Bardem	12	Herring Str	Har	NJ	7640
9	p09	Terence	Young	11	Broadway Av	Por	FL	34668
10	p10	Joel	Coen	30	East 10th S	New	NY	10003
11	p11	Ethan	Coen	30	East 10th S	New	NY	10003
12	p12	Spike	Jonze	20	Parva Ave	Los	CA	90027

The functional dependencies are the following:

pid → firstName, lastName, addressNumber, addressStreet, city, state, zipcode

---

ACTORS DATA

Next, the actors table looks like the following:

```
CREATE TABLE actors(
     aid char(3) NOT NULL,
     pid char(3) NOT NULL references people(pid),
     birthday char(10),
     hairColor text,
     eyeColor text,
     heightInches int,
     weightPounds int,
     sagAnniversaryDate char(10),
     primary key (aid)
     );

     INSERT INTO actors(aid, pid, birthday, hairColor, eyeColor,
heightInches, weightPounds, sagAnniversaryDate)
     VALUES ('a01', 'p01', '08/25/1930', 'grey', 'black', 74, 210,
            '10/12/1975'),
            ('a02', 'p02', '02/17/1981', 'brown', 'black', 69, 185,
             '12/15/2008'),
            ('a03', 'p03', '10/28/1974', 'brown', 'blue', 68, 175,
             '01/25/2001'),
             ('a04', 'p05', '11/11/1974', 'black', 'green', 72, 190,
             '05/21/1999'),
             ('a05', 'p06', '11/22/1967', 'grey', 'brown', 68, 192,
             '04/12/2005'),
             ('a06', 'p08', '03/01/1969', 'chestnut', 'black', 71, 188,
              '02/24/2007'),
             ('a07', 'p12', '10/22/1969', 'blonde', 'blue', 67, 140,
              '04/04/1994');
```

The previous SQL statements, when queried, produce the following table:

```
select *
from actors;
```

	aid character(3)	pid character(3)	birthday character(10)	haircolor text	eyecolor text	heightinches integer	weightpounds integer	saganniversarydate character(10)
1	a01	p01	08/25/1930	grey	black	74	210	10/12/1975
2	a02	p02	02/17/1981	brown	black	69	185	12/15/2008
3	a03	p03	10/28/1974	brown	blue	68	175	01/25/2001
4	a04	p05	11/11/1974	black	green	72	190	05/21/1999
5	a05	p06	11/22/1967	grey	brown	68	192	04/12/2005
6	a06	p08	03/01/1969	chestnu	black	71	188	02/24/2007
7	a07	p12	10/22/1969	blonde	blue	67	140	04/04/1994

The formal dependency of this table is the following:
aid → pid, birthday, hairColor, eyeColor, heightInches, weightPounds, sadAnniversaryDate

---

DIRECTORS DATA
The following statements will produce the directors data table:

```
CREATE TABLE directors(
     did char(3) NOT NULL,
     pid char(3) NOT NULL references people(pid),
     filmSchool text,
     dirGuildAnniversaryDate char(10),
     primary key (did)
     );

INSERT INTO directors(did, pid, filmSchool, dirGuildAnniversaryDate)
VALUES ('d01', 'p02', 'Columbia University', '09/12/2014'),
       ('d02', 'p04', 'University College London', '08/28/2008'),
       ('d03', 'p07', 'New York University', '02/24/2001' ),
       ('d04', 'p09', 'Univeristy of Cambridge', '12/14/1968'),
       ('d05', 'p10', 'New York University', '09/12/2006'),
       ('d06', 'p11', 'Princeton University', '09/12/2006') ,
       ('d07', 'p12', 'San Francisco Art Institute', '10/15/2006');
```

The output of the statements up above produce the following:

```
select *
from directors;
```

	did character(3)	pid character(3)	filmschool text	dirguildanniversarydate character(10)
1	d01	p02	Columbia University	09/12/2014
2	d02	p04	University College London	08/28/2008
3	d03	p07	New York University	02/24/2001
4	d04	p09	Univeristy of Cambridge	12/14/1968
5	d05	p10	New York University	09/12/2006
6	d06	p11	Princeton University	09/12/2006
7	d07	p12	San Francisco Art Institute	10/15/2006

The formal dependencies of the data are the following:

did → pid, filmschool, dirGuildAnniversaryDate

Movie Data

The following statements produced the movie table:

```
CREATE TABLE movies(
     mid char(3) NOT NULL,
     title text NOT NULL,
     yearOfRelease int,
     domesticBoxUSD int,
     foreignBoxUSD int,
     dvdBluSalesUSD int,
     primary key (mid)
     );

INSERT INTO movies (mid, title, yearOfRelease, domesticBoxUSD,
foreignBoxUSD, dvdBluSalesUSD)
VALUES ('m01', 'Dr. No', 1962, 16067035, 43432965, 10000),
       ('m02', 'Foxcatcher', 2014, 19000000, 2400000, 0),
       ('m03', 'Memento', 2001, 25544867, 14178229, 15000000),
       ('m04', 'No Country for Old Men', 2007, 74283625, 97343541,
        51800000),
       ('m05', 'Don Jon', 2013, 24477704, 5973052, 13000000),
       ('m06', 'Her', 2013, 25568251, 21783000, 12000000),
       ('m07', 'Whats Eating Gilbert Grape', 1993, 10032765, 1003453,
       125000);
```

The output of the statements above are as follows:

```
select *
from movies
```

	mid character(3)	title text	yearofrelease integer	domesticboxusd integer	foreignboxusd integer	dvdblusalesusd integer
1	m01	Dr. No	1962	16067035	43432965	10000
2	m02	Foxcatcher	2014	19000000	2400000	0
3	m03	Memento	2001	25544867	14178229	15000000
4	m04	No Country for Old Men	2007	74283625	97343541	51800000
5	m05	Don Jon	2013	24477704	5973052	13000000
6	m06	Her	2013	25568251	21783000	12000000
7	m07	Whats Eating Gilbert Grape	1993	10032765	1003453	125000

The functional dependencies of this table are as follows:

mid → title, yearOfRelease, domesticBoxOffice, foreignBoxUSD, dvdBluSales

---

Acts table

The following statements produce the ACTS table:

```
CREATE TABLE acts(
     aid char(3) references actors(aid),
     mid char(3) references movies(mid),
     primary key (aid, mid)
     );

INSERT INTO acts (aid, mid)
     VALUES ('a01', 'm01'),
            ('a02', 'm05'),
            ('a03', 'm06'),
            ('a04', 'm07'),
            ('a05', 'm02'),
            ('a06', 'm04'),
            ('a07', 'm06');
```

	aid character(3)	mid character(3)
1	a01	m01
2	a02	m05
3	a03	m06
4	a04	m07
5	a05	m02
6	a06	m04
7	a07	m06

The output of the previous statements is to the top-right.

The functional dependencies of this table are as follows:
aid, mid →

Directs Table

The following statements produce the DIRECTS table:

```
CREATE TABLE directs(
     did char(3) references directors(did),
     mid char(3) references movies(mid),
     primary key (did, mid)
     );

INSERT INTO directs (did, mid)
     VALUES ('d01', 'm05'),
            ('d02', 'm03'),
            ('d03', 'm02'),
            ('d04', 'm01'),
            ('d05', 'm04'),
            ('d06', 'm04'),
            ('d07', 'm06');


            select *
            from directs;
```

	did character(3)	mid character(3)
1	d01	m05
2	d02	m03
3	d03	m02
4	d04	m01
5	d05	m04
6	d06	m04
7	d07	m06

The output of the previous statements produce the following:

The functional dependencies of the table are as follows: did, mid →

The following query returns all the directors with whom actor “Sean Connery” has worked with.

```
select p.firstName, p.lastName
from people p, directors d1
where p.pid = d1.pid
  and d1.did = (select d2.did
                from directs d2, acts a1, movies m
                where a1.mid = m.mid
                and d2.mid = m.mid
                and a1.aid = (select a2.aid
                              from actors a2, people p
                              where a2.pid = p.pid
                              and p.firstName = 'Sean'
                              and p.lastName = 'Connery'
                              )
               );
```