**Final Project: Databases, Andrew Johnson**

**Outline**

I have created a database representing the relationship of people involved in making motion pictures. The entities included are actors, spouses, directors, and films. Relationships are marriages, actors acting in films, directors directing films, actors in religions, spouses in religions, and directors in religions.

The relationships are nothing technical and would be familiar to everyone. I thought it would be good to learn with a familiar example, although many people are acutely interested in celebrities.

**Database Outline in Words**

Actors have unique identifier ID, first name, last name, and birth date. Actors must have an ID, first name, and last name they cannot be null.

Spouses have unique identifier ID, first name, last name, and birth date. Spouses must have an ID, first name, and last name they cannot be null.

Directors have unique identifier ID, first name, last name, and birth date. Directors must have an ID, first name, and last name they cannot be null.

Films have unique film ID, title, starting budget, and projected release date. Films must have a title it cannot be null.

Religions have unique religion ID, name of religion, and date of founding. Religions must have a name, it cannot be null.

Actors will have 0 or 1 religion. They will also have 0 or 1 spouse as well. Actors can be in 1 or more films.

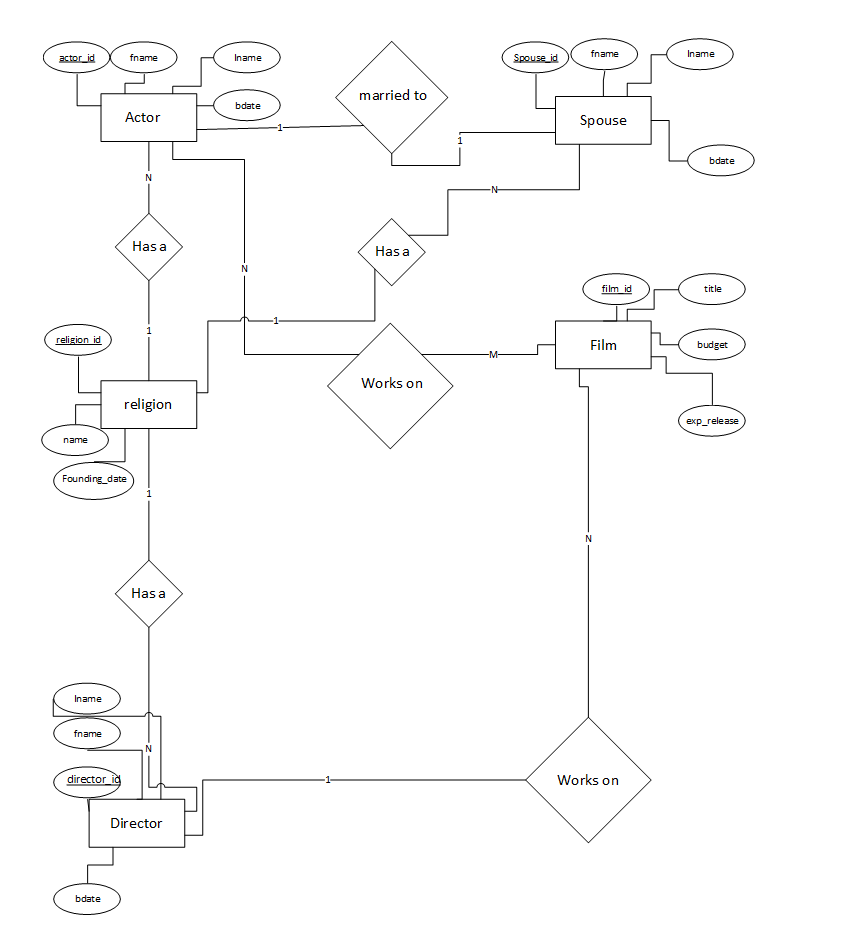
Directors will have 0 or more films that they are working on, directors can work on more than one film, but films can only have 0 or 1 directors. Directors will have 0 or 1 religion. Director spouses are not tracked.

Films will have 0 or 1 directors and 0 or more actors (we can have a film without actors here because we are not interested in voice actors, so animated films will have no actors, also films might not have a director in the planning stages).

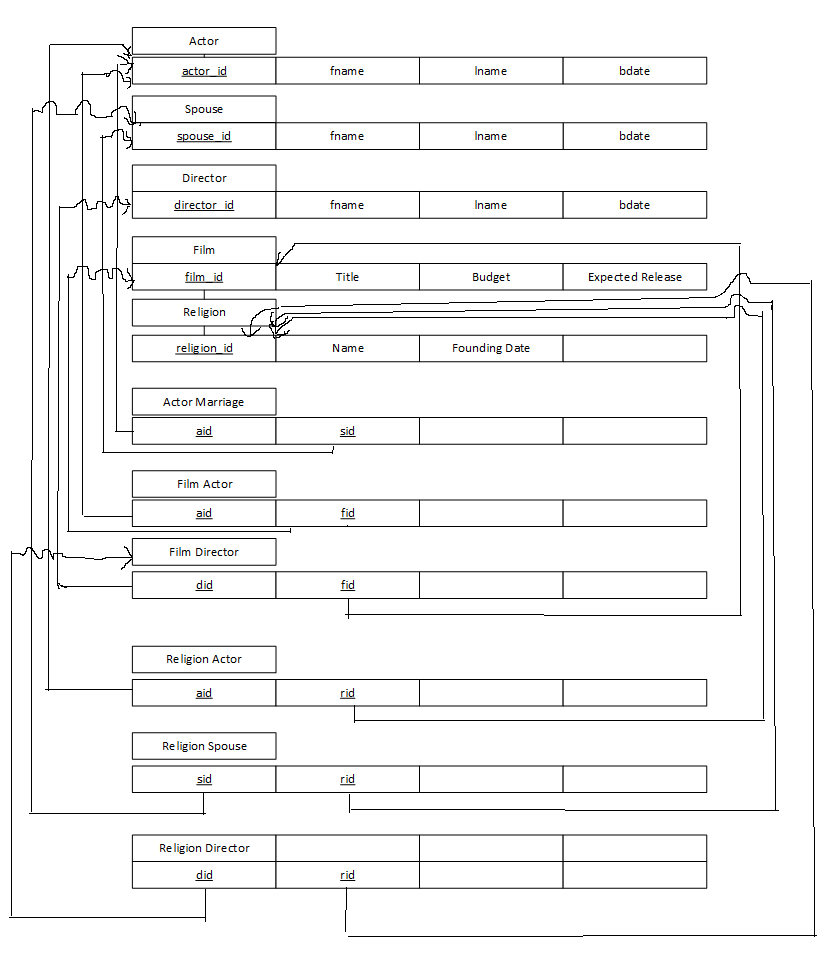
Spouses will have 1 actor that they are married to. No more no less. Spouses cannot be actors, this database tracks only non-acting spouses so there is no overlap. Spouses can have 0 or 1 religion.

Religions can have 0 or more members.

**ER Diagram**

****

**Schema**

****

I apologize for the crudity of this drawing, I was extremely careful but it is very detailed.

**Table Creation Queries**

CREATE TABLE actor(

actor\_id int(11) unsigned NOT NULL AUTO\_INCREMENT,

fname varchar(45) NOT NULL,

lname varchar(45) NOT NULL,

bdate DATE,

PRIMARY KEY (actor\_id)

)ENGINE=InnoDB

CREATE TABLE spouse(

spouse\_id int(11) unsigned NOT NULL AUTO\_INCREMENT,

fname varchar(45) NOT NULL,

lname varchar(45) NOT NULL,

bdate DATE,

PRIMARY KEY (spouse\_id)

)ENGINE=InnoDB

CREATE TABLE director(

director\_id int(11) unsigned NOT NULL AUTO\_INCREMENT,

fname varchar(45) NOT NULL,

lname varchar(45) NOT NULL,

bdate DATE,

PRIMARY KEY (director\_id)

)ENGINE=InnoDB

CREATE TABLE film(

film\_id int(11) unsigned NOT NULL AUTO\_INCREMENT,

title varchar(45) NOT NULL,

budget numeric(13,2),

exp\_release DATE,

PRIMARY KEY (film\_id)

)ENGINE=InnoDB

CREATE TABLE religion(

religion\_id int(11) unsigned NOT NULL AUTO\_INCREMENT,

name varchar(45) NOT NULL,

PRIMARY KEY (religion\_id),

foundingDate DATE

)ENGINE=InnoDB

CREATE TABLE actor\_marriage(

aid int(11) unsigned UNIQUE NOT NULL,

sid int(11) unsigned UNIQUE NOT NULL,

PRIMARY KEY (aid,sid),

CONSTRAINT FOREIGN KEY (aid) REFERENCES actor (actor\_id) ON DELETE CASCADE,

CONSTRAINT FOREIGN KEY (sid) REFERENCES spouse (spouse\_id) ON DELETE CASCADE

)ENGINE=InnoDB

CREATE TABLE film\_actor(

aid int(11) unsigned NOT NULL,

fid int(11) unsigned NOT NULL,

PRIMARY KEY (aid,fid),

CONSTRAINT FOREIGN KEY (aid) REFERENCES actor (actor\_id) ON DELETE CASCADE,

CONSTRAINT FOREIGN KEY (fid) REFERENCES film (film\_id) ON DELETE CASCADE

)ENGINE InnoDB

CREATE TABLE film\_director(

did int(11) unsigned NOT NULL,

fid int(11) unsigned UNIQUE NOT NULL,

PRIMARY KEY (did,fid),

CONSTRAINT FOREIGN KEY (did) REFERENCES director (director\_id) ON DELETE CASCADE,

CONSTRAINT FOREIGN KEY (fid) REFERENCES film (film\_id) ON DELETE CASCADE

)ENGINE InnoDB

CREATE TABLE religion\_actor(

rid int(11) unsigned NOT NULL,

aid int(11) unsigned UNIQUE NOT NULL,

PRIMARY KEY (rid,aid),

CONSTRAINT FOREIGN KEY (rid) REFERENCES religion (religion\_id) ON DELETE CASCADE,

CONSTRAINT FOREIGN KEY (aid) REFERENCES actor (actor\_id) ON DELETE CASCADE

)ENGINE InnoDB

CREATE TABLE religion\_spouse(

rid int(11) unsigned NOT NULL,

sid int(11) unsigned UNIQUE NOT NULL,

PRIMARY KEY (rid,sid),

CONSTRAINT FOREIGN KEY (rid) REFERENCES religion (religion\_id) ON DELETE CASCADE,

CONSTRAINT FOREIGN KEY (sid) REFERENCES spouse (spouse\_id) ON DELETE CASCADE

)ENGINE InnoDB

CREATE TABLE religion\_director(

rid int(11) unsigned NOT NULL,

did int(11) unsigned UNIQUE NOT NULL,

PRIMARY KEY (rid,did),

CONSTRAINT FOREIGN KEY (rid) REFERENCES religion (religion\_id) ON DELETE CASCADE,

CONSTRAINT FOREIGN KEY (did) REFERENCES director (director\_id) ON DELETE CASCADE

)ENGINE InnoDB

**General Use Queries**

**Used many times to show what table currently contains so user can run other operations.**

SELECT actor.fname, actor.lname, actor.bdate FROM actor

SELECT spouse.fname, spouse.lname, spouse.bdate FROM spouse

SELECT director.fname, director.lname, director.bdate FROM director

SELECT film.title, film.budget, film.exp\_release FROM film

SELECT religion.name, religion.foundingDate FROM religion

SELECT a.fname, a.lname, f.title FROM film\_actor fa INNER JOIN film f ON f.film\_id = fa.fid INNER JOIN actor a ON fa.aid=a.actor\_id

SELECT d.fname, d.lname, f.title FROM film\_director fd INNER JOIN film f ON f.film\_id = fd.fid INNER JOIN director d ON fd.did=d.director\_id

SELECT a.fname, a.lname, s.fname, s.lname FROM actor\_marriage am INNER JOIN actor a ON a.actor\_id = am.aid INNER JOIN spouse s ON am.sid=s.spouse\_id

SELECT d.fname, d.lname, r.name FROM religion\_director rd INNER JOIN director d ON d.director\_id = rd.did INNER JOIN religion r ON rd.rid=r.religion\_id

SELECT s.fname, s.lname, r.name FROM religion\_spouse rs INNER JOIN spouse s ON s.spouse\_id = rs.sid INNER JOIN religion r ON rs.rid=r.religion\_id

**Used many times to create spouse drop down for selection**

SELECT actor\_id, fname, lname FROM actor

SELECT spouse\_id, fname, lname FROM spouse

SELECT director\_id, fname, lname FROM director

SELECT film\_id, title FROM film

SELECT religion\_id, name FROM religion

**Queries used in Filters, adds, deletes etc.**

INSERT INTO actor (fname, lname, bdate) VALUES ([firstname input],[lastname input],[birtdate input])

INSERT INTO director (fname, lname, bdate) VALUES ([firstname input],[lastname input],[birtdate input])

INSERT INTO film (title, budget, exp\_release) VALUES ([title input],[budget input],[releaseDate input])

INSERT INTO spouse (fname, lname, bdate) VALUES ([firstname input],[lastname input],[birtdate input])

DELETE FROM actor WHERE actor\_id[actor id input]

DELETE FROM spouse WHERE spouse\_id=[spouse id input]

DELETE FROM director WHERE director\_id=[director id input]

DELETE FROM film WHERE film\_id=[film id input]

INSERT INTO religion (name, foundingDate) VALUES ([name input, founding date input?)

DELETE FROM religion WHERE religion\_id=[religion ID input]

INSERT INTO film\_actor (aid,fid) VALUES ([actor id input],[film id input])

INSERT INTO actor\_marriage (aid,sid) VALUES ([actor id input], [film id input

SELECT a.fname, a.lname, s.fname, s.lname FROM actor a INNER JOIN actor\_marriage am ON a.actor\_id=am.aid INNER JOIN spouse s ON s.spouse\_id=am.sid WHERE a.actor\_id=[actor id input]

SELECT a.fname, a.lname, f.title FROM actor a INNER JOIN film\_actor fa ON a.actor\_id = fa.aid INNER JOIN film f ON fa.fid=f.film\_id WHERE a.actor\_id = [actor id input]

SELECT a.fname, a.lname, f.title FROM actor a INNER JOIN film\_actor fa ON fa.aid=a.actor\_id INNER JOIN film f ON f.film\_id=fa.fid WHERE f.film\_id = [film id input]

INSERT INTO film\_director (did,fid) VALUES ([director id input], [film id input])

INSERT INTO religion\_director (rid,did) VALUES ([religion id input], [director id input])

SELECT d.fname, d.lname, f.title FROM director d INNER JOIN film\_director fd ON fd.did=d.director\_id INNER JOIN film f ON f.film\_id=fd.fid WHERE f.film\_id = [film id input]

INSERT INTO religion\_actor (rid,aid) VALUES ([religion id input],[actor id input])

INSERT INTO religion\_spouse (rid,sid) VALUES ([religion id input], [spouse id input])

DELETE FROM film\_director WHERE did = [director id input]

DELETE FROM actor\_marriage WHERE aid = [actor id input]

DELETE FROM religion\_actor WHERE aid = [actor id input]

DELETE FROM religion\_director WHERE did = [director id input]

DELETE FROM religion\_spouse WHERE sid = [spouse id input]

SELECT r.name, a.fname, a.lname, s.fname, s.lname, d.fname, d.lname FROM religion\_actor ra

INNER JOIN religion\_director rd ON ra.rid=rd.rid INNER JOIN religion\_spouse rs ON rd.rid=rs.rid INNER JOIN actor a ON ra.aid=a.actor\_id INNER JOIN spouse s ON s.spouse\_id=rs.sid INNER JOIN director d ON rd.did=d.director\_id INNER JOIN religion r ON r.religion\_id=rd.rid WHERE r.religion\_id = [religion id input]

DELETE FROM film\_actor WHERE aid=[actor id input]&&fid=[film id input]