



--Trevor Pirone-- --The SQL script is also on GitHub as well as the query if it is too hard to read here.--

--Normalization 2 Lab #8--

--CMPT 308--

--Creating and connecting to the database. If the tables exist already, drop them.--

DROP TABLE IF EXISTS Persons, Actors, Directors, Movies, Movie\_Directors, Movie\_Actors;

--Else, proceed with the creation of Persons table--

```

CREATE TABLE Persons (
    person_id INT NOT NULL,
    f_name VARCHAR(36) NOT NULL,
    l_name VARCHAR(36) NOT NULL,
    address VARCHAR(36),
    PRIMARY KEY(person_id)
);
  
```

--Creating the Actors table.--

```
CREATE TABLE Actors (  
    actor_id INT REFERENCES Persons(person_id),  
    birth_date DATE,  
    hair_color VARCHAR(16),  
    eye_color VARCHAR(16),  
    height_inches INT,  
    weight_pounds INT,  
    actorsGuildAnniversaryDate DATE,  
    PRIMARY KEY(actor_id)  
);
```

--Creating the Directors table.--

```
CREATE TABLE Directors (  
    director_id INT REFERENCES Persons(person_id),  
    school_name VARCHAR(72),  
    directorsGuildAnniversaryDate DATE,  
    PRIMARY KEY(director_id)  
);
```

--Creating the Movies table.--

```
CREATE TABLE Movies (  
    movie_id INT NOT NULL,  
    movie_name VARCHAR(100) NOT NULL,  
    year INT,  
    domesticBoxOfficeSalesUSD INT,  
    foreignBoxOfficeSalesUSD INT,  
    mediaOpticSalesUSD INT,  
    PRIMARY KEY(movie_id)  
);
```

--Creating the Movie Directors table.--

```
CREATE TABLE Movie_Directors (  
    movie_id INT REFERENCES Movies(movie_id),  
    director_id INT REFERENCES Directors(director_id)  
);
```

--Creating the Movie Actors table.--

```
CREATE TABLE Movie_Actors (  
    movie_id INT REFERENCES Movies(movie_id),  
    actor_id INT REFERENCES Actors(actor_id)  
);
```

--Insert values into Person table.--

INSERT INTO Persons (person\_id, f\_name, l\_name, address) VALUES

(1, 'Roger', 'Moore', 'The Labousuer Residence'),

(2, 'He Who Must', 'Not Be Named', '666th St., Pirone Town, Pirone'),

(3, 'Sean', 'Conway', '3399 South Road'),

(4, 'Shane', 'Connery', 'The City of Angels, CA'),

(5, 'Shawn', 'Canary', '911 Burgess Road'),

(6, 'Shawne', 'Connor E.', 'Everywhere, Universe'),

(7, 'Shaun', 'Canorli', '6 Short Lane'),

(8, 'Not Sean', 'Connery', '6 Long Street'),

(9, 'Sean', 'Not Connery', 'Planet Eath'),

(10, 'Normal', 'Director', 'The Basement');

--Insert values in Actors table.--

INSERT INTO Actors (actor\_id, birth\_date, hair\_color, eye\_color, height\_inches, weight\_pounds, actorsGuildAnniversaryDate) VALUES

(1, '1927-10-14', 'Brown', 'Blue', '73', '175', '1945-08-26'),

(2, '1930-08-25', 'Brown', 'Brown', '74', '190', '1949-06-16');

--Insert values into Directors table.--

```
INSERT INTO Directors (director_id, school_name, directorsGuildAnniversaryDate) VALUES
(3, 'Marist College', '2005-02-24'),
(4, 'Vassar College', '2002-01-14'),
(5, 'Iona College', '2006-09-29'),
(6, 'Providence College', '2001-07-11'),
(7, 'University of Buffalo', '2003-06-06'),
(8, 'UCLA', '2010-12-13'),
(9, 'NYU', '2012-03-20'),
(10, 'High School', '2017-11-11');
```

--Insert values into Movies table.--

```
INSERT INTO Movies(movie_id, movie_name, year, domesticBoxOfficeSalesUSD,
foreignBoxOfficeSalesUSD, mediaOpticSalesUSD) VALUES
(1, 'Amazing Movie', 1997, 1000087, 576843, 12124),
(2, 'The Life of Trevor', 2017, 9999999, 99999999, 99999999),
(3, 'Alan is Love, Alan is Life', 2015, 1237890, 9873210, 5555555),
(4, 'Duy is Life, Duy is Love', 2016, 8584239, 1920409, 19293095);
```

--Insert values into Movie Directors table.--

INSERT INTO Movie\_Directors (movie\_id, director\_id) VALUES

(1, 10),

(2, 3),

(2, 4),

(2, 7),

(3, 5),

(3, 9),

(4, 6),

(4, 8);

--Insert values into Movie Actors table.--

INSERT INTO Movie\_Actors (movie\_id, actor\_id) VALUES

(1, 2),

(2, 1),

(3, 1),

(4, 1),

(4, 2);

```
--Query for all directors who worked with Roger Moore.--  
SELECT person_id, f_name, l_name  
FROM Persons  
WHERE person_id IN (SELECT person_id  
                    FROM Persons, Movie_Directors  
                    WHERE Persons.person_id = Movie_Directors.director_id  
                    AND Movie_Directors.director_id IN (SELECT Movie_Directors.director_id  
                                                         FROM Movie_Directors, Movie_Actors  
                                                         WHERE Movie_Directors.movie_id = Movie_Actors.movie_id  
                                                         AND Movie_Actors.movie_id IN (SELECT Movie_Actors.movie_id  
                                                                 FROM Movie_Actors, Persons  
                                                                 WHERE Movie_Actors.actor_id = Persons.person_id  
                                                                 AND Persons.person_id IN (SELECT Persons.person_id  
                                                                     FROM Persons  
                                                                     WHERE Persons.f_name = 'Roger'  
                                                                     AND Persons.l_name = 'Moore')  
                                                                 )  
                                                         )  
                    )  
  
ORDER BY person id;
```

## Functional Dependencies

person\_id  $\rightarrow$  f\_name

person\_id  $\rightarrow$  l\_name

person\_id  $\rightarrow$  address

actor\_id  $\rightarrow$  birth\_date

actor\_id  $\rightarrow$  hair\_color

actor\_id  $\rightarrow$  eye\_color

actor\_id  $\rightarrow$  height\_inches

actor\_id  $\rightarrow$  weight\_pounds

actor\_id  $\rightarrow$  actorsGuildAnniversaryDate

director\_id  $\rightarrow$  school\_name

director\_id  $\rightarrow$  directorsGuildAnniversaryDate

movie\_id  $\rightarrow$  movie\_name

movie\_id  $\rightarrow$  year

movie\_id  $\rightarrow$  domesticBoxOfficeSalesUSD

movie\_id  $\rightarrow$  foreignBoxOfficeSalesUSD

movie\_id  $\rightarrow$  opticalMeidaSalesUSD



