Joseph Rowell

```
CREATE table IF NOT EXISTS person (

person_Id                               SERIAL NOT NULL UNIQUE,
first_Name                               VARCHAR(50)    NOT     NULL,
last_Name                                VARCHAR(50)    NOT     NULL,
street_Address                           VARCHAR(255)  NOT     NULL,

Primary key (person_Id)
);
```

Functional dependencies:
person_Id →first_Name, last_Name, street_Address

```
/*CREATE TYPE hair_Color AS ENUM ('black', 'brown', 'blonde', 'white', 'grey', 'red', 'auburn', 'other');
CREATE TYPE eye_Color AS ENUM ('Amber', 'Blue', 'Brown', 'Gray', 'Green', 'Hazel', 'Red', 'Violet');

The commented code did not work.*/

Create table IF NOT EXISTS actors (

person_Id INTEGER NOT NULL,
birth_Date     DATE  NOT NULL,
hairC               VARCHAR(18),
eyeC                VARCHAR(18),
heightInches      NUMERIC CONSTRAINT personHeight CHECK (heightInches > 0),
weightLbs         NUMERIC CONSTRAINT personWeight CHECK (weightLbs > 0),
SAG_Date    DATE  NOT NULL,
Primary key (actors_Id),
actors_Id int REFERENCES           person(person_Id)

) inherits(person);
```

Functional dependencies:
person_Id → hair_Color, eye_Color, heightInches, weightLbs, SAG_Date

```
Create table IF NOT EXISTS directors (
d_Id int REFERENCES person(person_Id),
film_School       VARCHAR(100) NULL,
DGA_Date          DATE     NOT NULL,

Primary key (d_Id)
) inherits(person);
```

Functional dependencies
Person_Id → film_School, DGA_Date

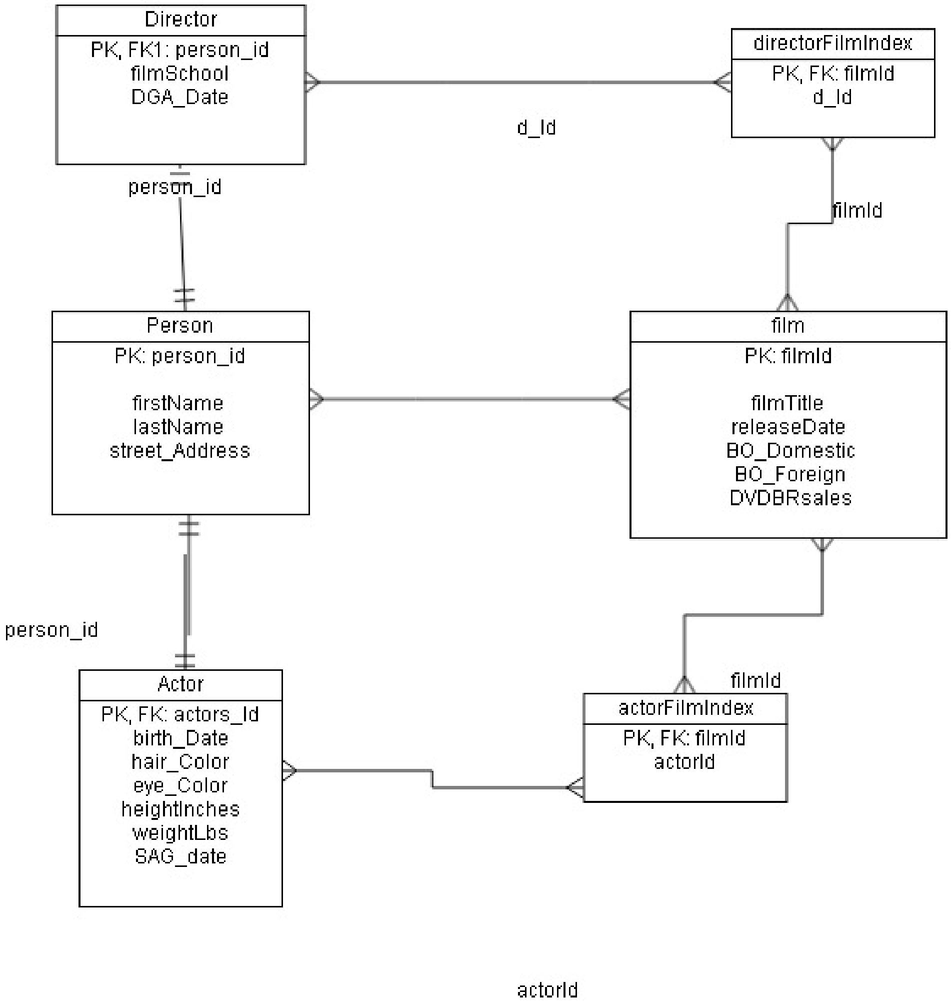
```
Create table IF NOT EXISTS films (

film_Id          SERIAL NOT NULL,
filmTitle        VARCHAR(100) NOT NULL,
releaseDate      DATE    NOT NULL,
BO_Domestic      MONEY   NOT NULL,
BO_Foreign       MONEY   NOT NULL,
DVDBRsales       MONEY   NOT NULL,
PRIMARY KEY (film_Id)
);
```

Functional Dependencies: filmId-> filmTitle, releasedate, BO_Domestic, BO_Foreign, DVDBRsales

```
Create table actorFilmIndex (
film_Id int NOT NULL references films,
actors_Id int NOT NULL REFERENCES actors (actors_Id),
PRIMARY KEY (film_Id, actors_Id)
)

Create table directorFilmIndex (
film_Id int NOT NULL references films,
d_Id int NOT NULL references directors (d_Id),
PRIMARY KEY (film_Id, d_Id)
)

//INSERTS

Insert into person (first_Name, last_Name, street_Address)
VALUES
('Sean', 'Connery', '7 No Street'),
('George', 'Lazenby', '8 Secret Street'),
('Roger', 'Moore', '9 SpyParadise Street'),
('Terence', 'Young', ' 11 Youngs blvd'),
('Guy', 'Hamilton', '5 Ham Street'),
('Lewis', 'Gilbert', '44 Epic Street'),
('Peter', 'Hunt', '17 London Way');


INSERT INTO actors(agent_Id, birth_Date, hairC, eyeC, heightInches, weightLbs, SGA_Date)
VALUES
(1, '1930-08-25', 'Black', 'Brown', '74', '200', '1954-04-18'),
(2, '1939-09-05', 'Black', 'Brown', '74', '180', '1965-09-17'),
(3, '1930-10-14', 'Black', 'Brown', '73', '195', '1945-04-18');
```

```
INSERT INTO directors(d_Id,first_Name,last_Name, film_School, DGA_Date)
VALUES
(4, ‘John’, ‘Glen', Paris, '1940-04-10'),
(5, ‘Guy’, ‘Hamilton’, 'None', '1952-06-30'),
(6, ‘Martin’, ‘Campbell’, 'None', '1966-07-04'),
(7, ‘Peter’, ‘Hunt’, 'Norway', '1940-07-18');

INSERT INTO films (filmTitle, releasedate, BO_Domestic, BO_Foreign, DVDBRsales)
VALUES
('Dr. No', 1962, 16.1, 43.5, 30.2),
('From Russia with Love', 1963, 24.8, 54.1, 41.5),
('Goldfinger', 1964, 51.1, 73.8, 79.1),
('Thunderball', 1965, 63.6, 77.6, 92.1),
('You Only Live Twice', 1967, 43.1, 68.5, 69.5),
('On Her Majesty/’s Secret Service', 1969, 22.8, 59.2, 32.4),
('Live and Let Die', 1973, 35.4, 91.0, 56.9);


INSERT INTO actorFilmIndex (filmId, actor_Id)
VALUES
(1, 1),
(2, 1),
(3, 1),
(4, 2),
(5, 5),
(6, 2),
(7, 3);

INSERT INTO directorFilmIndex (filmId, d_Id)
VALUES
(1, 4),
(2, 7),
(3, 5),
(4, 4),
(5, 6),
(6, 7),
(7, 5);

SELECT first_Name, last_Name FROM person
WHERE person_Id IN (
 SELECT person_Id FROM directors
 WHERE film_Id IN (
 SELECT film_Id FROM actorFilmIndex
 WHERE person_Id IN (
 SELECT person_Id FROM Person
 WHERE first_Name, last_Name = 'Sean Connery'
 )
 )
);
```

//For some reason, this does not work. I'll fix it later.


Director
PK, FK1: person_id
filmSchool
DGA_Date
directorFilmIndex
PK, FK: filmId
d_Id
d_Id
person_id
filmId
Person
PK: person_id
firstName
lastName
street_Address
film
PK: filmId
filmTitle
releaseDate
BO_Domestic
BO_Foreign
DVDBRsales
person_id
filmId
Actor
PK, FK: actors_Id
birth_Date
hair_Color
eye_Color
heightInches
weightLbs
SAG_date
actorFilmIndex
PK, FK: filmId
actorId
actorId