



```

--drop tables if exist--
DROP TABLE IF EXISTS ActorFilmography;
DROP TABLE IF EXISTS DirectorFilmography;
DROP TABLE IF EXISTS Actors;
DROP TABLE IF EXISTS Movies;
DROP TABLE IF EXISTS Directors;
DROP TABLE IF EXISTS Address;
DROP TABLE IF EXISTS FilmSchool;

```

```

--Address--
CREATE TABLE Address
( AddressID integer not null,
  StreetName varchar(255) not null,
  City varchar(255) not null,
  State varchar(255) not null,
  ZipCode integer not null,
  Country varchar(255) not null,
  primary key(AddressID)
);

```

```

--Actors--
CREATE TABLE Actors
( ActorID integer not null,
  FirstName varchar(255) not null,
  LastName varchar(255) not null,
  AddressID integer references Address(AddressID),
  BirthDate date,
  HairColor varchar(25),
  EyeColor varchar(25),
  HeightInches integer,
  WeightPounds integer,

```

```

    ActorsGuildAnniversary date,
    primary key(ActorID)
);

--Movies--
CREATE TABLE Movies
( MovieID      integer not null,
  MovieName    varchar(255) not null,
  YearReleased integer,
  BoxOfficeSalesDomestic integer,
  BoxOfficeSalesForeign integer,
  DVDSales     integer,
  BluRaySales integer,
  primary key(MovieID)
);

--FilmSchool--
CREATE TABLE FilmSchool
( SchoolID      integer not null,
  SchoolName    varchar(255) not null,
  ParentUniversity varchar(255),
  StreetName    varchar(255),
  City          varchar(50),
  State         varchar(50),
  Country       varchar(255),
  primary key(SchoolID)
);

--Directors--
CREATE TABLE Directors
( DirectorID integer not null,
  FirstName  varchar(255) not null,
  LastName   varchar(255) not null,
  AddressID  integer references Address(AddressID),
  SchoolID   integer references FilmSchool(SchoolID),
  DirectorsGuildAnniversary date,
  primary key(DirectorID)
);

--ActorsFilmography--
CREATE TABLE ActorFilmography
( MovieID      integer not null references Movies(MovieID),
  ActorID      integer not null references Actors(ActorID),
  CharacterName varchar(255) not null,
  primary key(MovieID, ActorID, CharacterName)
);

--DirectorFilmography--
CREATE TABLE DirectorFilmography
( MovieID      integer not null references Movies(MovieID),
  DirectorID   integer not null references Directors(DirectorID),
  primary key(MovieID, DirectorID)
);

--InsertIntoAddress--
INSERT INTO Address( AddressID, StreetName, City, State, ZipCode, Country)
VALUES (1, '1 BestActorEver Lane', 'Hollywood', 'California', 91601,
        'United States of America');

INSERT INTO Address( AddressID, StreetName, City, State, ZipCode, Country)
VALUES (2, '2 BestActorEver Lane', 'Hollywood', 'California', 91601,

```

```

        'United States of America');

INSERT INTO Address( AddressID, StreetName, City, State, ZipCode, Country)
VALUES (3, '1 DirectorsLane', 'Hollywood', 'California', 91601,
        'United States of America');

INSERT INTO Address( AddressID, StreetName, City, State, ZipCode, Country)
VALUES (4, '2 DirectorsLane', 'Hollywood', 'California', 91601,
        'United States of America');

--InsertIntoActors--
INSERT INTO Actors( ActorID, FirstName, LastName, AddressID, BirthDate,
                    HairColor, EyeColor, HeightInches, WeightPounds,
                    ActorsGuildAnniversary)
VALUES(1, 'Leonardo', 'Dicaprio', 1, '1974-11-11', 'Brown', 'Dreamy Blue', 72, 165,
'1990-1-1');

INSERT INTO Actors( ActorID, FirstName, LastName, AddressID, BirthDate,
                    HairColor, EyeColor, HeightInches, WeightPounds,
                    ActorsGuildAnniversary)
VALUES(2, 'Christian', 'Bale', 2, '1974-1-30', 'Brown', 'Brown', 72, 150, '1986-1-
1');

--InsertIntoMovies--
INSERT INTO Movies( MovieID, MovieName, YearReleased, BoxOfficeSalesDomestic,
                    BoxOfficeSalesForeign,DVDSales, BluRaySales)
VALUES(1, 'American Hustle', 2014, 45000000, 20000000, 2, 800000);

INSERT INTO Movies( MovieID, MovieName, YearReleased, BoxOfficeSalesDomestic,
                    BoxOfficeSalesForeign,DVDSales, BluRaySales)
VALUES(2, 'The Great Gatsby', 2013, 100000000, 22000000, 9000, 999999);

--InsertIntoFilmSchool--
INSERT INTO FilmSchool( SchoolID, SchoolName, ParentUniversity, StreetName, City,
                        State, Country)
VALUES(1, 'School of Film and Animation', 'Rochester Institute of Technology',
        '1 Lomb Memorial Drive', 'Rochester', 'New York', 'United States of America');

INSERT INTO FilmSchool( SchoolID, SchoolName, ParentUniversity, StreetName, City,
                        State, Country)
VALUES(2, 'Jack and Vivian Hanson Film Institute', 'University of Arizona',
        '550 E Van Buren Street', 'Phoenix', 'Arizona', 'United States of America');

--InsertIntoDirectors--
INSERT INTO Directors( DirectorID, FirstName, LastName, AddressID,
                       SchoolID, DirectorsGuildAnniversary)
VALUES(1, 'Baz', 'Luhrmann', 3, 1, '1666-1-1');

INSERT INTO Directors( DirectorID, FirstName, LastName, AddressID,
                       SchoolID, DirectorsGuildAnniversary)
VALUES(2, 'David', 'Russel', 4, 2, '1776-1-1');

--InsertIntoActorsFilmography--
INSERT INTO ActorFilmography( MovieID, ActorID, CharacterName)
VALUES (1, 2, 'Irving Rosenfeld');

INSERT INTO ActorFilmography( MovieID, ActorID, CharacterName)
VALUES (2, 1, 'Jay Gatsby');

--InsertIntoDirectorsFilmography--
INSERT INTO DirectorFilmography( MovieID, DirectorID)
VALUES (1, 1);

```

```
INSERT INTO DirectorFilmography( MovieID, DirectorID)
VALUES (2, 2);
```

Functional Dependencies:

Actors:

ActorID → FirstName, LastName, AddressID, BirthDate, HairColor, EyeColor, HeightInches, WeightPounds, ActorsGuildAnniversary

Address:

AddressID → StreetName, City, State, ZipCode, Country

Directors:

DirectorID → FirstName, LastName, AddressID, SchoolID, DirectorsGuildAnniversary

Movies:

MovieID → MovieName, YearReleased, BoxOfficeSalesDomestic, BoxOfficeSalesForeign, DVDSales, BluRaySales

FilmSchool

SchoolID → SchoolName, ParentUniversity, StreetName, City, Country

****This should work but it doesn't. Can't figure out why.****

```
select distinct Directors.FirstName, Directors.LastName
from Directors,
    DirectorFilmography
where DirectorFilmography.MovieID in (select MovieID
                                     from Movies
                                     where MovieID in (select MovieID
                                                         from ActorFilmography
                                                         where ActorID in (select ActorID
                                                                               from Actors
                                                                               where (FirstName = 'Sean')
                                                                               and (LastName = 'Connery')
                                                                               )
                                                         )
                                     )
);
```