CREATE DATABASE Basic\_SQL\_ReTest\_LamPV2

GO

USE Basic\_SQL\_ReTest\_LamPV2

GO

CREATE TABLE actor

(

actor\_id INT PRIMARY KEY IDENTITY,

first\_name NVARCHAR(50),

last\_name NVARCHAR(50),

last\_update DATE DEFAULT GETDATE(),

)

CREATE TABLE category

(

category\_id INT PRIMARY KEY IDENTITY,

name NVARCHAR(50) NOT NULL UNIQUE,

last\_update DATE DEFAULT GETDATE()

)

CREATE TABLE film

(

film\_id INT PRIMARY KEY IDENTITY,

title NVARCHAR(50) NOT NULL UNIQUE,

description\_ NVARCHAR(50),

release\_year DATE DEFAULT GETDATE(),

leng\_ INT,

rating INT,

special\_features NVARCHAR(50),

last\_update DATE DEFAULT GETDATE()

)

CREATE TABLE film\_category

(

film\_id INT FOREIGN KEY(film\_id) REFERENCES film(film\_id),

category\_id INT FOREIGN KEY(category\_id) REFERENCES category(category\_id),

last\_update DATE DEFAULT GETDATE(),

PRIMARY KEY(film\_id, category\_id)

)

CREATE TABLE film\_actor

(

actor\_id INT FOREIGN KEY(actor\_id) REFERENCES actor(actor\_id),

film\_id INT FOREIGN KEY(film\_id) REFERENCES film(film\_id),

last\_update DATE DEFAULT GETDATE(),

PRIMARY KEY(actor\_id, film\_id)

)

INSERT INTO film(title) VALUES(N'Lady Bird')

INSERT INTO film(title) VALUES(N'Get Out')

INSERT INTO film(title) VALUES(N'Coco')

INSERT INTO film(title) VALUES(N'Wonder Woman')

INSERT INTO category(name) VALUES(N'Animation')

INSERT INTO category(name) VALUES(N'Comedy')

INSERT INTO category(name) VALUES(N'Drama')

INSERT INTO category(name) VALUES(N'Kids')

INSERT INTO actor(first\_name, last\_name) VALUES('Angelina', 'Jolie')

INSERT INTO actor(first\_name, last\_name) VALUES('Jennifer', 'Lawrence')

INSERT INTO actor(first\_name, last\_name) VALUES('Alain', 'Delon')

INSERT INTO actor(first\_name, last\_name) VALUES('Megan', 'Fox')

INSERT INTO film\_actor(actor\_id, film\_id) VALUES(1, 3)

INSERT INTO film\_actor(actor\_id, film\_id) VALUES(2, 3)

INSERT INTO film\_actor(actor\_id, film\_id) VALUES(4, 3)

INSERT INTO film\_actor(actor\_id, film\_id) VALUES(1, 1)

INSERT INTO film\_actor(actor\_id, film\_id) VALUES(3, 2)

INSERT INTO film\_category(film\_id, category\_id) VALUES(1, 1)

INSERT INTO film\_category(film\_id, category\_id) VALUES(1, 2)

INSERT INTO film\_category(film\_id, category\_id) VALUES(3, 3)

INSERT INTO film\_category(film\_id, category\_id) VALUES(4, 2)

INSERT INTO film\_category(film\_id, category\_id) VALUES(2, 2)

--2.Write an query list all actor first name and last name acting in the film title.

--Example, input film title ‘Coco’ return list actors ‘Angelina Jolie’, ‘Jennifer Lawrence’, ‘Megan Fox’

--Nhập tiêu đề phim in ra diễn viên phim đó

SELECT CONCAT(actor.first\_name,' ' ,actor.last\_name)

FROM film INNER JOIN film\_actor

ON film.film\_id = film\_actor.film\_id INNER JOIN actor

ON film\_actor.actor\_id = actor.actor\_id

WHERE film.title = 'Coco'

--3.Write:

-- A query which given a category name parameter that returns total film for a specific

--category name. Example input category name ‘Comedy’ return 3 films.

--Nhập vào loại phim và in ra số phim của loại đó

SELECT film.title

FROM film INNER JOIN film\_category

ON film.film\_id = film\_category.film\_id INNER JOIN category

ON film\_category.category\_id = category.category\_id

WHERE category.name = 'Comedy'

--4.Add constraint for Rating field is not a number in range [1-5] or Length field is negative number.

-- Thay đổi dữ liệu cho rating từ 1-5 và leng không âm

ALTER TABLE film ADD CONSTRAINT rating CHECK(rating >= 1 AND rating <= 5)

ALTER TABLE film ADD CONSTRAINT leng\_ CHECK(leng\_ > 0)

--5.Write an query find films with the most actors.

--Tìm phim nhiều diễn viên nhất

SELECT \* FROM film

WHERE film.film\_id = (

SELECT TOP 1 WITH TIES film\_actor.film\_id FROM film\_actor

GROUP BY film\_actor.film\_id

ORDER BY COUNT(film\_actor.actor\_id) DESC)