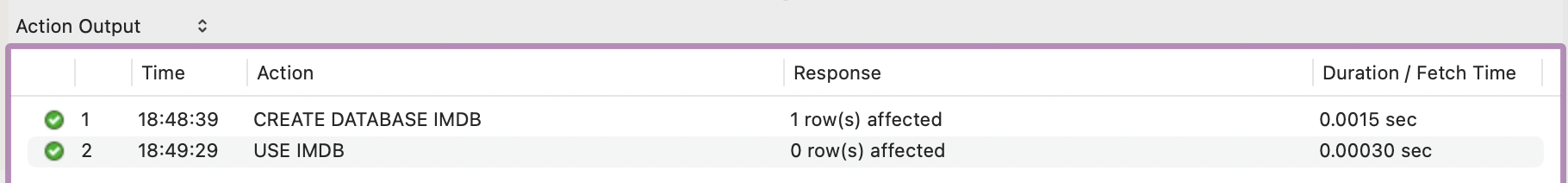
**MySQL : Creation of database IMDB and tables along with PRIMARY KEY, FOREIGN KEY creation and insertion of data into tables.**

CREATE DATABASE IMDB;

USE IMDB;



CREATE TABLE movie(

**movie\_id INT PRIMARY KEY NOT NULL,**

title VARCHAR(100) NOT NULL,

director VARCHAR(100) NOT NULL,

release\_year YEAR

);

INSERT INTO movie(movie\_id,title,director,release\_year) VALUES

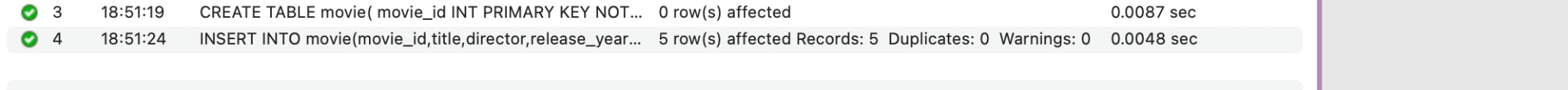
(101,'Dangal','Nitesh Tiwari',2016),

(102,'Baahubali2','Rajamouli',2017),

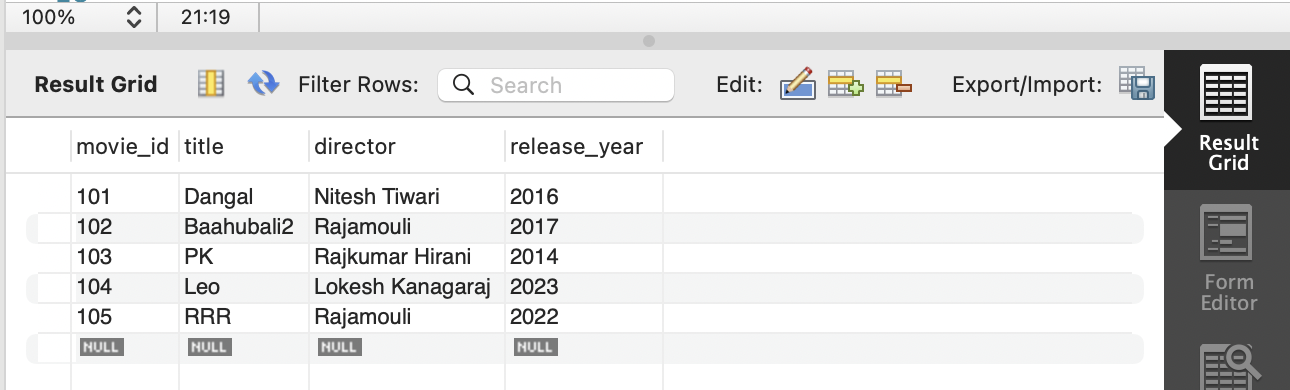
(103,'PK','Rajkumar Hirani',2014),

(104,'Leo','Lokesh Kanagaraj',2023),

(105,'RRR','Rajamouli',2022);



SELECT \* FROM movie;



**#movie\_id is used as a FOREIGN KEY**

CREATE TABLE media(

**media\_id INT PRIMARY KEY AUTO\_INCREMENT,**

movie\_id INT,

media\_type VARCHAR(200),

media\_url VARCHAR(200),

**FOREIGN KEY(movie\_id) REFERENCES movie(movie\_id)**

);

INSERT INTO media(movie\_id,media\_type,media\_url) VALUES

(101,'Dangal.mp4','https://dangal.mp4'),

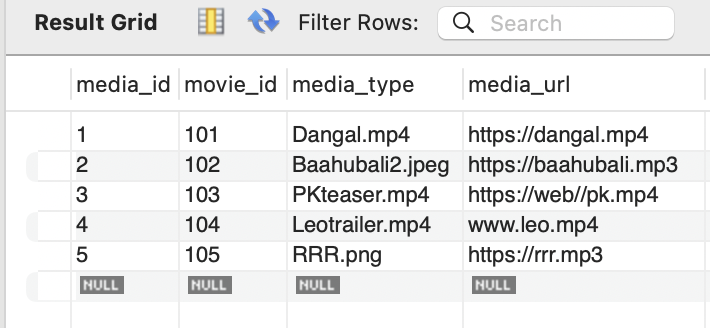
(102,'Baahubali2.jpeg','https://baahubali.mp3'),

(103,'PKteaser.mp4','https://web//pk.mp4'),

(104,'Leotrailer.mp4','www.leo.mp4'),

(105,'RRR.png','<https://rrr.mp3>');

SELECT \* FROM media;



CREATE TABLE genre(

**genre\_id INT PRIMARY KEY AUTO\_INCREMENT,**

movie\_id INT,

movie\_genre VARCHAR(100),

**FOREIGN KEY(movie\_id) REFERENCES movie(movie\_id)**

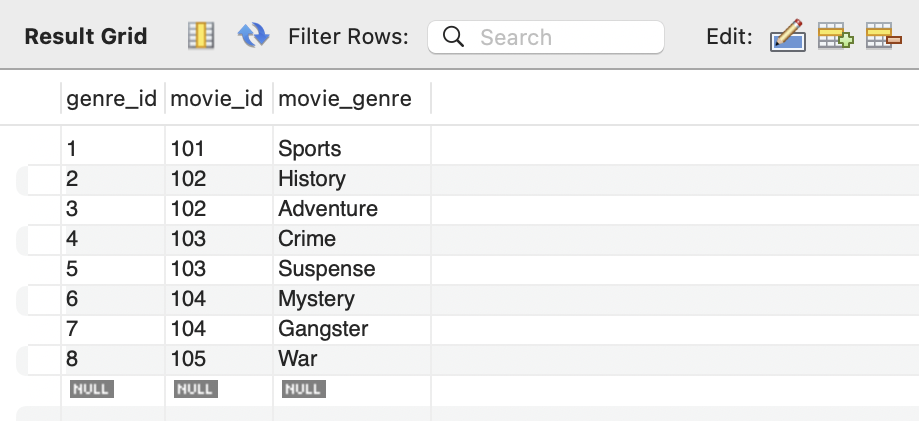
);

INSERT INTO genre(movie\_id,movie\_genre) VALUES

(101,'Sports'),(102,'History'),(102,'Adventure'),(103,'Crime'),(103,'Suspense'),

(104,'Mystery'),(104,'Gangster'),(105,'War');

SELECT \* FROM genre;



CREATE TABLE user(

**user\_id INT PRIMARY KEY NOT NULL,**

**movie\_id INT,**

influencer VARCHAR(50),

people VARCHAR(50),

film\_critics VARCHAR(50),

journalists VARCHAR(50),

**FOREIGN KEY(movie\_id) REFERENCES movie(movie\_id)**

);

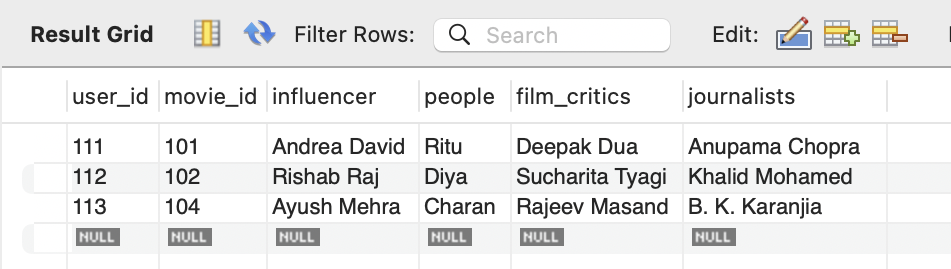
INSERT INTO user(user\_id,movie\_id,influencer,people,film\_critics,journalists) VALUES

(111,101,'Andrea David','Ritu','Deepak Dua','Anupama Chopra'),

(112,102,'Rishab Raj','Diya','Sucharita Tyagi','Khalid Mohamed'),

(113,104,'Ayush Mehra','Charan','Rajeev Masand','B. K. Karanjia');

SELECT \* FROM user;



CREATE TABLE reviews(

**review\_id INT PRIMARY KEY AUTO\_INCREMENT,**

**user\_id INT,**

**FOREIGN KEY(user\_id) REFERENCES user(user\_id),**

comments VARCHAR(200)

);

INSERT INTO reviews(user\_id,comments) VALUES

(111,'DANGAL is a masterpiece'),

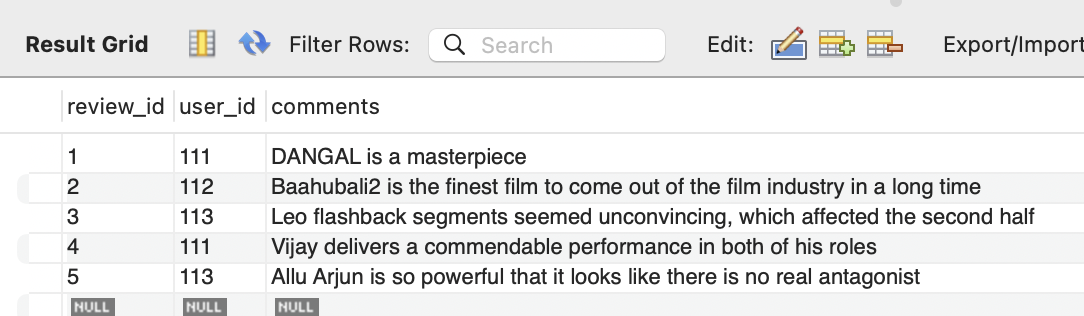
(112,'Baahubali2 is the finest film to come out of the film industry in a long time'),

(113,'Leo flashback segments seemed unconvincing, which affected the second half'),

(111,'Vijay delivers a commendable performance in both of his roles'),

(113,'Allu Arjun is so powerful that it looks like there is no real antagonist');

SELECT \* FROM reviews;



CREATE TABLE artist(

**artist\_id INT PRIMARY KEY,**

**movie\_id INT,**

artist\_name VARCHAR(100),

skills VARCHAR(100),

**FOREIGN KEY(movie\_id) REFERENCES movie(movie\_id)**

);

INSERT INTO artist(artist\_id,movie\_id,artist\_name,skills) VALUES

(201,101,'Aamir Khan','Martial Arts'),

(202,101,'Fatima Sana','Dancer'),

(203,102,'Prabhas','Fighter'),

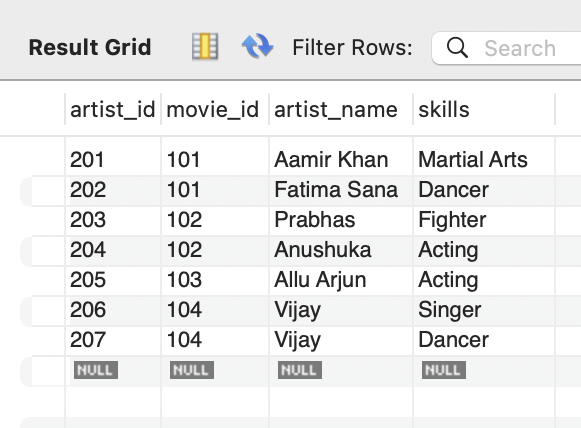
(204,102,'Anushuka','Acting'),

(205,103,'Allu Arjun','Acting'),

(206,104,'Vijay','Singer'),

(207,104,'Vijay','Dancer');

SELECT \* FROM artist;



CREATE TABLE roles(

artist\_id INT,

artist\_role VARCHAR(100),

**FOREIGN KEY(artist\_id) REFERENCES artist(artist\_id)**

);

INSERT INTO roles(artist\_id,artist\_role) VALUES

(201,'Trainer'),

(202,'Student'),

(203,'Hero'),

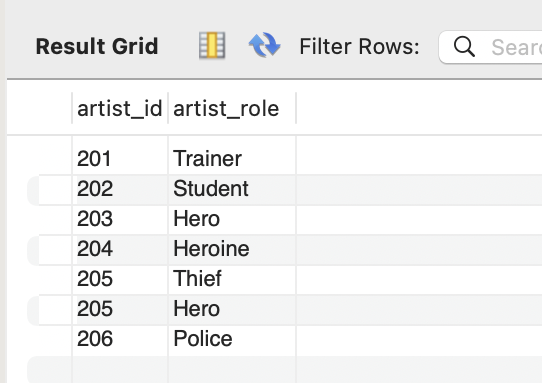
(204,'Heroine'),

(205,'Thief'),

(205,'Hero'),

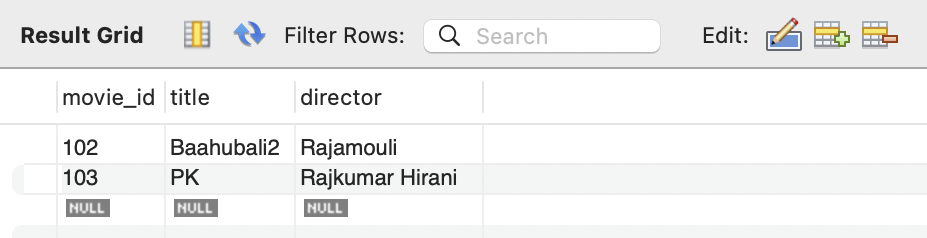
(206,'Police');

SELECT \* FROM roles;

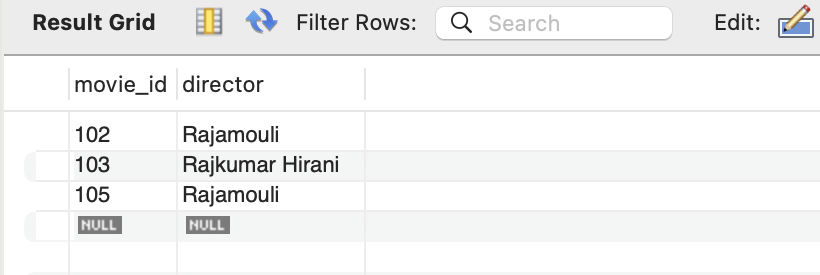


**SQL Queries performed on the created tables:**

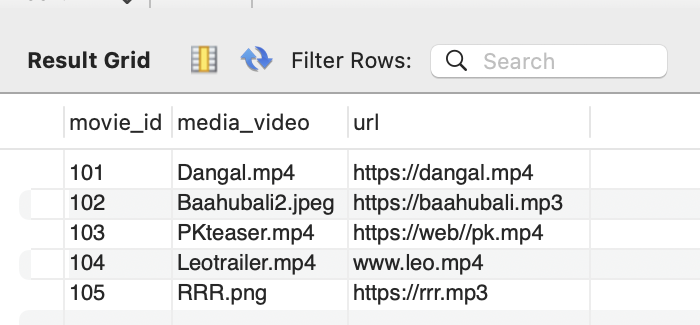
SELECT movie\_id,title,director from movie where release\_year **IN (2014,2017);**



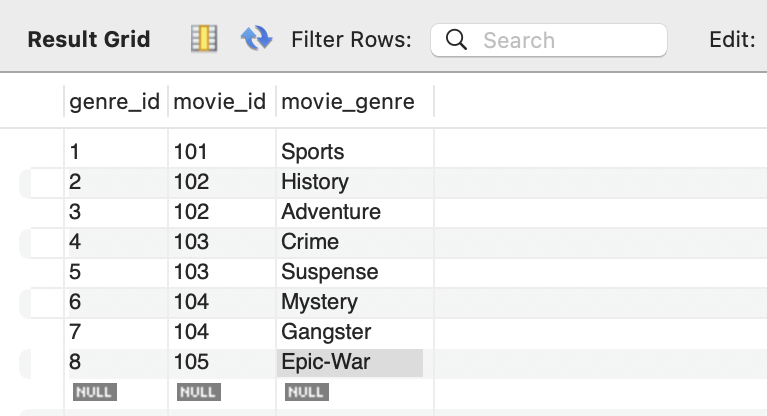
SELECT movie\_id,director from movie where director LIKE 'R%';



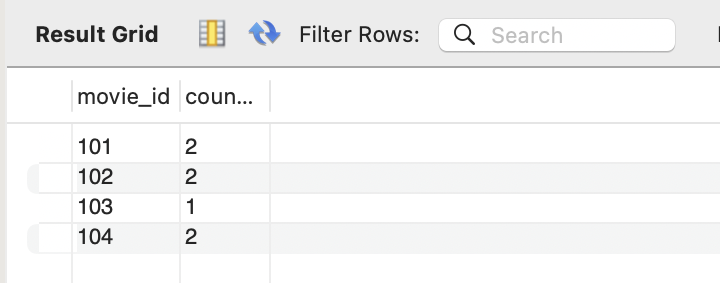
SELECT movie\_id,**media\_type as media\_video, media\_url as url** FROM media;



**UPDATE genre set movie\_genre='Epic-War' where movie\_id=105;**

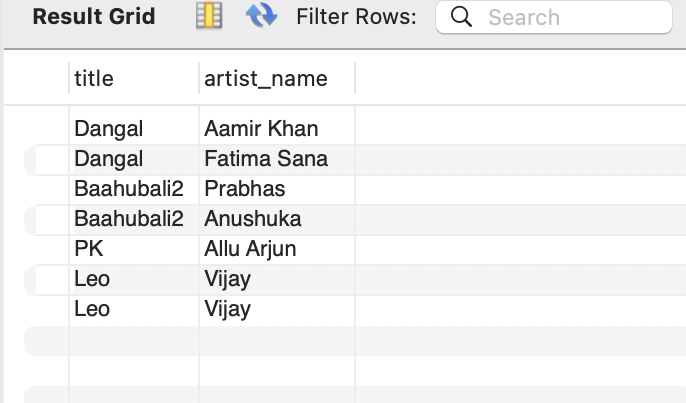


SELECT movie\_id**,COUNT(\*)** from artist **GROUP BY** movie\_id;



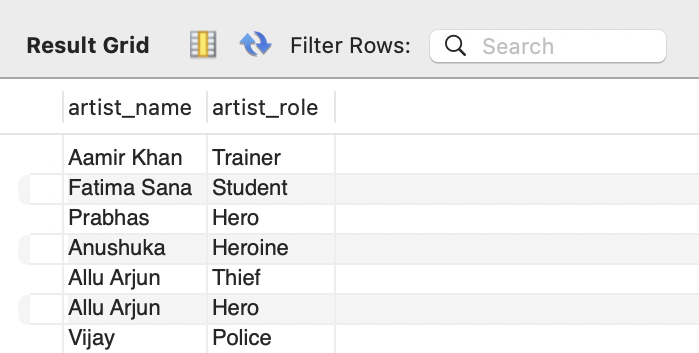
Select title,artist\_name from movie

**INNER JOIN** artist where movie.movie\_id=artist.movie\_id;

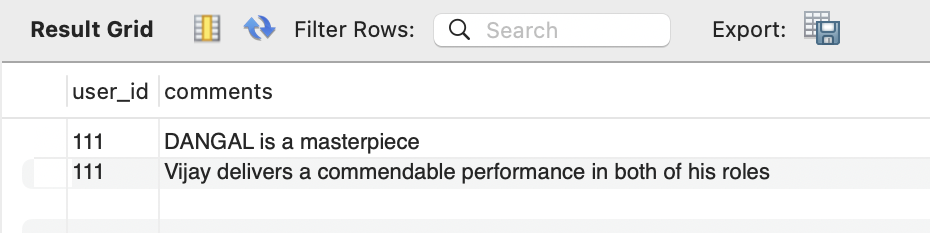


SELECT artist\_name, artist\_role FROM artist

**RIGHT JOIN** roles on artist.artist\_id=roles.artist\_id;



SELECT user\_id,comments FROM reviews where **user\_id= (SELECT user\_id from user LIMIT 1);**



DROP TABLE roles; #deletes the entire table

DROP DATABASE IMDB; #deletes the entire database