# 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;

Action O	utput \$			
	Time	Action	Response	Duration / Fetch Time
<b>0</b> 1	18:48:39	CREATE DATABASE IMDB	1 row(s) affected	0.0015 sec
<b>②</b> 2	18:49:29	USE IMDB	0 row(s) affected	0.00030 sec

### 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),

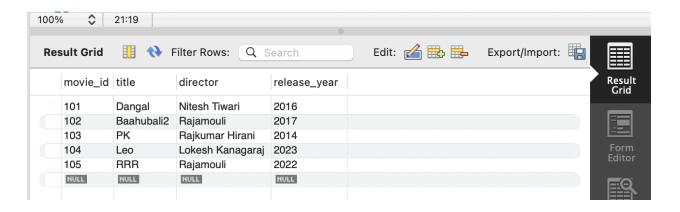
(103, 'PK', 'Rajkumar Hirani', 2014),

(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;

esult Grid	<b>III **</b>	Filter Rows:	Q Search
media_ic	movie_id	media_type	media_url
1	101	Dangal.mp4	https://dangal.mp4
2	102	Baahubali2.jpeg	https://baahubali.mp3
3	103	PKteaser.mp4	https://web//pk.mp4
4	104	Leotrailer.mp4	www.leo.mp4
5	105	RRR.png	https://rrr.mp3
HULL	NULL	NULL	NULL

```
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;

Result Grid	<b>#</b> **	Filter Rows:	Q Search	Edit: 🚄 🎛 🏪
genre_id	movie_id	movie_genre		
1	101	Sports		
2	102	History		
3	102	Adventure		
4	103	Crime		
5	103	Suspense		
6	104	Mystery		
7	104	Gangster		
8	105	War		
NULL	NULL	NULL		

#### CREATE TABLE user(

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

influencer VARCHAR(50),

people VARCHAR(50),

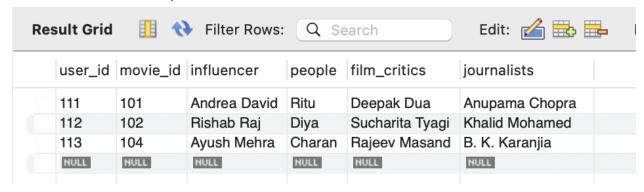
film\_critics VARCHAR(50),

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'),

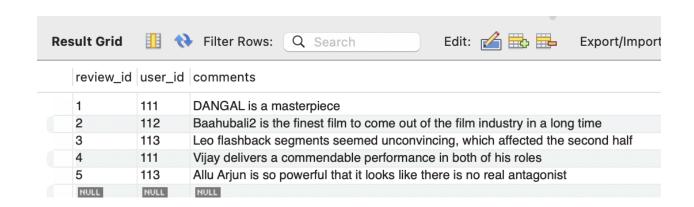
(112, 'Baahubali2 is the finest film to come out of the film industry in a long time'),

(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;

sult Grid	<b>III 43</b>	Filter Rows	: Q Sea
artist_id	movie_id	artist_name	skills
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
NULL	NULL	NULL	HULL

```
CREATE TABLE roles(
artist_id INT,
artist_role VARCHAR(100),
FOREIGN KEY(artist_id) REFERENCES artist(artist_id)
);
```

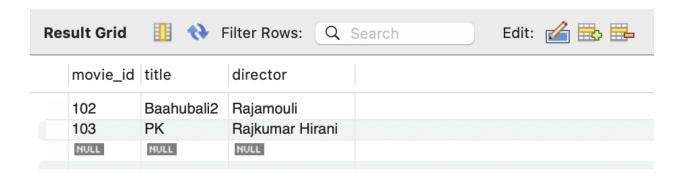
```
INSERT INTO roles(artist_id,artist_role) VALUES (201,'Trainer'), (202,'Student'), (203,'Hero'), (204,'Heroine'), (205,'Thief'), (205,'Hero'), (206,'Police');
```

## SELECT \* FROM roles;

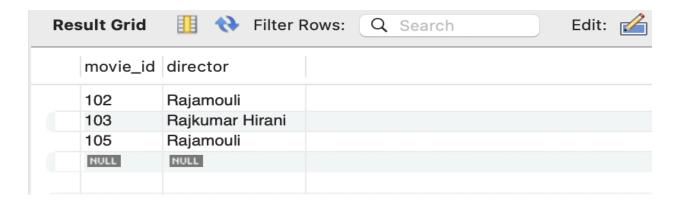
Result Grid	<b>**</b>	Filter Rows:	Q Searc
artist_id	artist_role		
201	Trainer		
202	Student		
203	Hero		
204	Heroine		
205	Thief		
205	Hero		
206	Police		

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

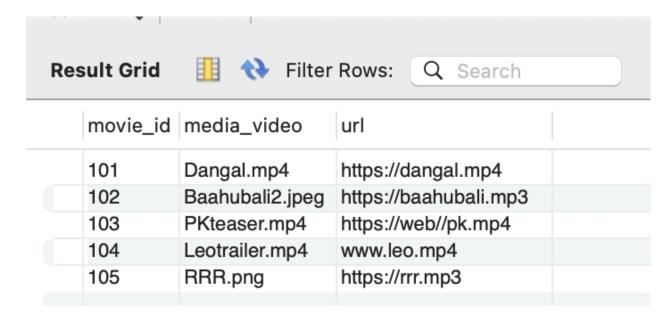
SELECT movie id, title, director from movie where release year IN (2014,2017);



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;

Res	sult Grid	<b>***</b>	Filter Rows:	Q Search	Edit:
	genre_id	movie_id	movie_genre		
	1	101	Sports		
	2	102	History		
	3	102	Adventure		
	4	103	Crime		
	5	103	Suspense		
	6	104	Mystery		
	7	104	Gangster		
	8	105	Epic-War		
	NULL	NULL	NULL		

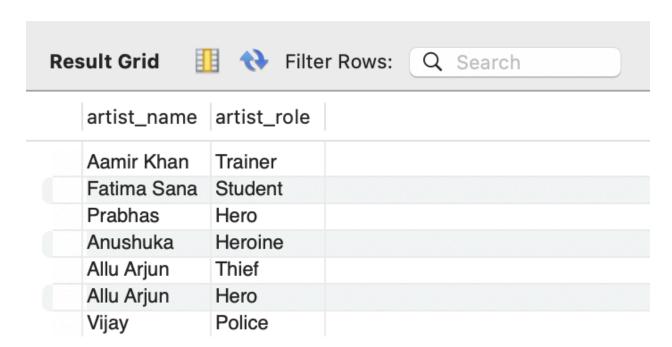
# SELECT movie\_id,COUNT(\*) from artist GROUP BY movie\_id;

Result Grid	Filter Rows: Q Search
movie_id	coun
101	2
102	2
103	1
104	2

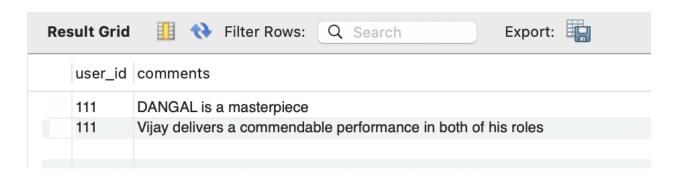
# Select title,artist\_name from movie INNER JOIN artist where movie.movie\_id=artist.movie\_id;

title	artist_name
Dangal	Aamir Khan
Dangal	Fatima Sana
Baahubali2	Prabhas
Baahubali2	Anushuka
PK	Allu Arjun
Leo	Vijay
Leo	Vijay

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