# Task-16

In this file contains screenshots of the MySQL shell commands:

using MySQL, design a database whose name is IMDB. create proper MySQL tables, primary key, foreign key, add data into the MySQL tables do the following given below :-

1.movie should have multiple media (video or image)

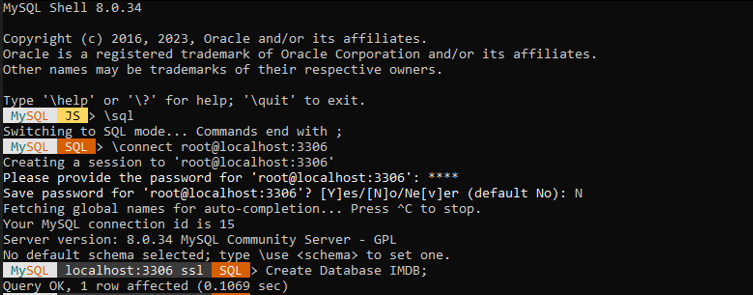
2.movie can belong to multiple genre

3.movie can have multiple reviews and reviews can belongs to user

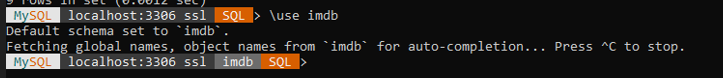
4.artist can have multiple skills

5.artist can perform multiple roles in a single film

-- Create IMDB database



--USE IMDB:



--create Table to store information about movie:

CREATE TABLE Movie (movie\_id INT PRIMARY KEY,title VARCHAR(70),release\_date DATE);



-- Add data for movie:

--INSERT INTO Movie (movie\_id, title, release\_date) VALUES:

A black background with white text

Description automatically generated

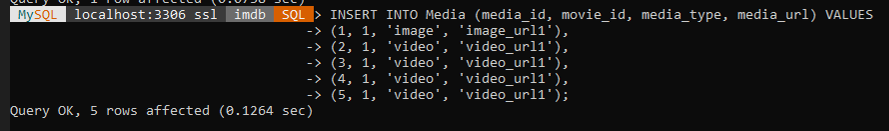
-- Table to store media related to movies (images or videos):

CREATE TABLE Media (media\_id INT PRIMARY KEY,movie\_id INT,media\_type ENUM('image', 'video'),media\_url VARCHAR(100),FOREIGN KEY (movie\_id) REFERENCES Movie(movie\_id));



-- Add media for movies:

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

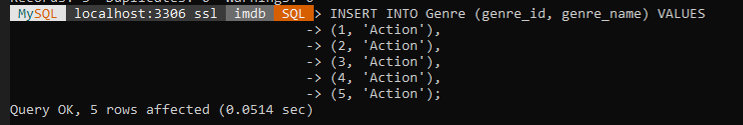


-- Table to store information about genres:

CREATE TABLE Genre (genre\_id INT PRIMARY KEY,genre\_name VARCHAR(50));



-- Add genres for movies:



-- Table to associate movies with genres:

CREATE TABLE MovieGenre (movie\_id INT,genre\_id INT,PRIMARY KEY (movie\_id, genre\_id),FOREIGN KEY (movie\_id) REFERENCES Movie(movie\_id),FOREIGN KEY (genre\_id) REFERENCES Genre(genre\_id));



-- Associate movies with genres:

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

A screen shot of a computer

Description automatically generated

-- Table to store information about reviews:



-- Add reviews for movies:

INSERT INTO Review (review\_id, movie\_id, user\_id, rating, review\_text) VALUES

A screen shot of a computer program

Description automatically generated

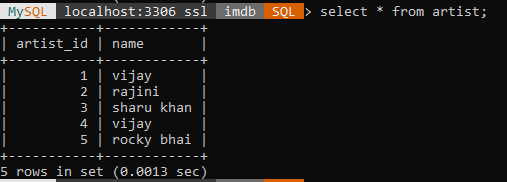
-- Table to store information about artists:

CREATE TABLE Artist (artist\_id INT PRIMARY KEY,name VARCHAR(20));



INSERT INTO Artist (artist\_id,name) VALUES

Use select \* from table\_name to get inserted values:



-- Table to store information about skills:

CREATE TABLE Skill (skill\_id INT PRIMARY KEY,skill\_name VARCHAR(50));



INSERT INTO SKILL (skill\_id,skill\_name) VALUES (1,'Fantastic comic timing'), (2,'High Self Esteem'), (3,'Life Skills from a Struggler'), (4,'Fantastic comic timing'), (5,'A hardworking actor'); 

-- Table to associate artists with skills:

CREATE TABLE ArtistSkill (artist\_id INT,skill\_id INT,PRIMARY KEY (artist\_id, skill\_id),FOREIGN KEY (artist\_id) REFERENCES Artist(artist\_id),FOREIGN KEY (skill\_id) REFERENCES Skill(skill\_id));



INSERT INTO ArtistSkill (artist\_id,skill\_id) VALUES A black screen with white text

Description automatically generated

-- Table to store information about roles in movies:

CREATE TABLE Role (role\_id INT PRIMARY KEY,role\_name VARCHAR(50));



INSERT INTO Role (role\_id,role\_name) VALUES (1,'Leo das'),(2,'muthuvel pandian'),(3,'pathaan'),(4,'veera raghavan'),(5,'rocky'); 

-- Table to associate artists with roles in movies:

CREATE TABLE ArtistRole (artist\_id INT,role\_id INT,movie\_id INT,PRIMARY KEY (artist\_id, role\_id, movie\_id),FOREIGN KEY (artist\_id) REFERENCES Artist(artist\_id),FOREIGN KEY (role\_id) REFERENCES Role(role\_id),FOREIGN KEY (movie\_id) REFERENCES Movie(movie\_id));

