Database and Tables creation:

🡪DB name is IMDB create table using my sql  
  
CREATE TABLE Genre (

genre\_id INT AUTO\_INCREMENT PRIMARY KEY,

genre\_name VARCHAR(50) NOT NULL

);

CREATE TABLE Movie (

movie\_id INT AUTO\_INCREMENT PRIMARY KEY,

movie\_title VARCHAR(100) NOT NULL,

release\_year YEAR,

duration\_minutes INT,

-- Other attributes of movie like director, producer, etc. can be added here

);

CREATE TABLE Movie\_Genre (

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)

);

CREATE TABLE Media (

media\_id INT AUTO\_INCREMENT PRIMARY KEY,

media\_type ENUM('video', 'image') NOT NULL,

media\_url VARCHAR(255) NOT NULL,

movie\_id INT,

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

);

CREATE TABLE Artist (

artist\_id INT AUTO\_INCREMENT PRIMARY KEY,

artist\_name VARCHAR(100) NOT NULL

-- Other artist details like birth date, nationality, etc. can be added here

);

CREATE TABLE Artist\_Skill (

artist\_id INT,

skill VARCHAR(50) NOT NULL,

PRIMARY KEY (artist\_id, skill),

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

);

CREATE TABLE Movie\_Artist (

movie\_id INT,

artist\_id INT,

role VARCHAR(50) NOT NULL,

PRIMARY KEY (movie\_id, artist\_id),

FOREIGN KEY (movie\_id) REFERENCES Movie(movie\_id),

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

);

CREATE TABLE User (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

-- Other user details like email, password, etc. can be added here

);

CREATE TABLE Review (

review\_id INT AUTO\_INCREMENT PRIMARY KEY,

movie\_id INT,

user\_id INT,

review\_text TEXT,

rating DECIMAL(3, 1), -- Assuming rating out of 10

review\_date DATE,

FOREIGN KEY (movie\_id) REFERENCES Movie(movie\_id),

FOREIGN KEY (user\_id) REFERENCES User(user\_id)

);

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

SELECT movie.movie\_id, movie.movie\_name

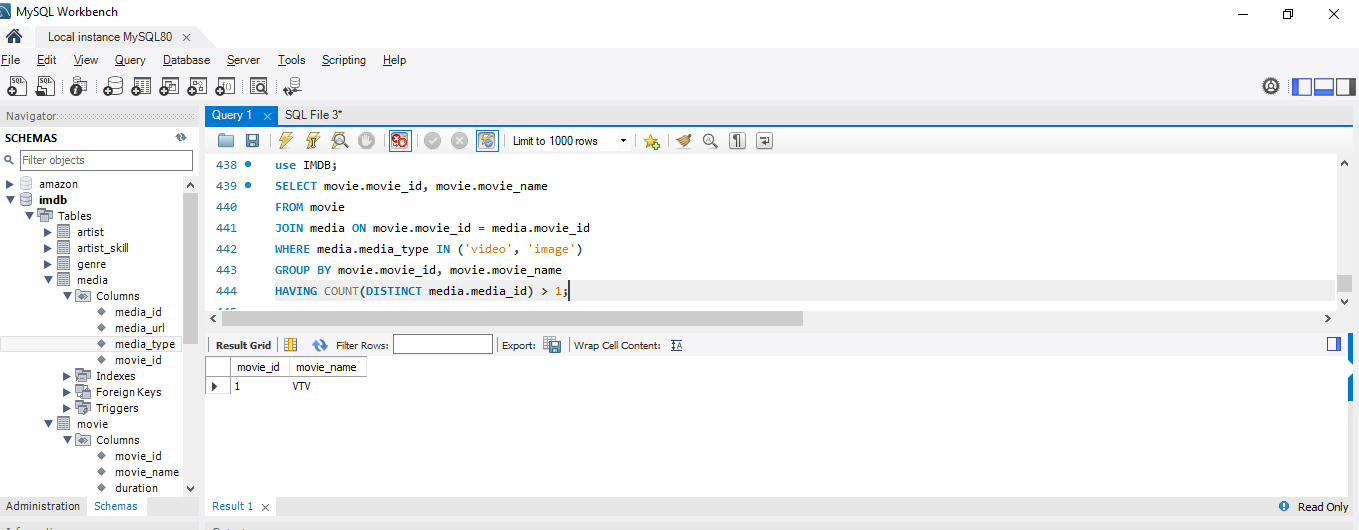
FROM movie

JOIN media ON movie.movie\_id = media.movie\_id

WHERE media.media\_type IN ('video', 'image')

GROUP BY movie.movie\_id, movie.movie\_name

HAVING COUNT(DISTINCT media.media\_id) > 1;



2.Movie can belongs to multiple genre

SELECT movie.movie\_id,movie.movie\_name,group\_concat(distinct genre.genre\_name) as genres

from movie

JOIN media on movie.movie\_id=media.movie\_id

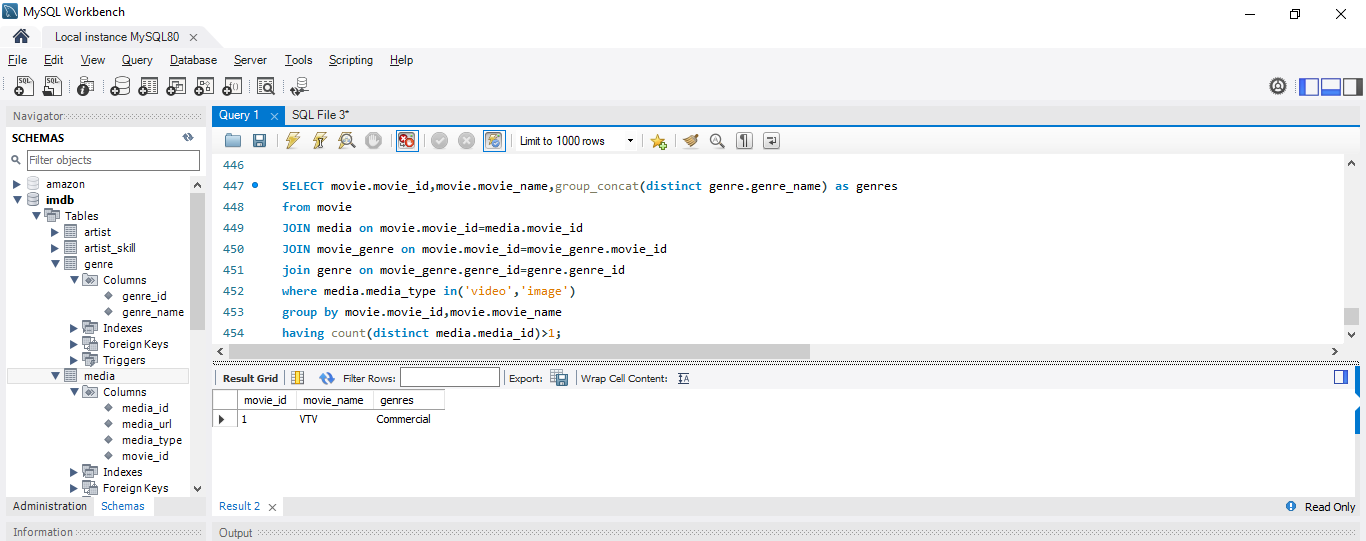
JOIN movie\_genre on movie.movie\_id=movie\_genre.movie\_id

join genre on movie\_genre.genre\_id=genre.genre\_id

where media.media\_type in('video','image')

group by movie.movie\_id,movie.movie\_name

having count(distinct media.media\_id)>1;



3.Movie can have multiple reviews and Review can belongs to a user

SELECT movie.movie\_id, movie.movie\_name,

GROUP\_CONCAT(DISTINCT genre.genre\_name) AS genres,

COUNT(DISTINCT media.media\_id) AS media\_count,

COUNT(DISTINCT review.review\_id) AS review\_count,

GROUP\_CONCAT(DISTINCT CONCAT(user.user\_name) ORDER BY review.review\_id SEPARATOR ', ') AS reviewers

FROM movie

JOIN media ON movie.movie\_id = media.movie\_id

JOIN movie\_genre ON movie.movie\_id = movie\_genre.movie\_id

JOIN genre ON movie\_genre.genre\_id = genre.genre\_id

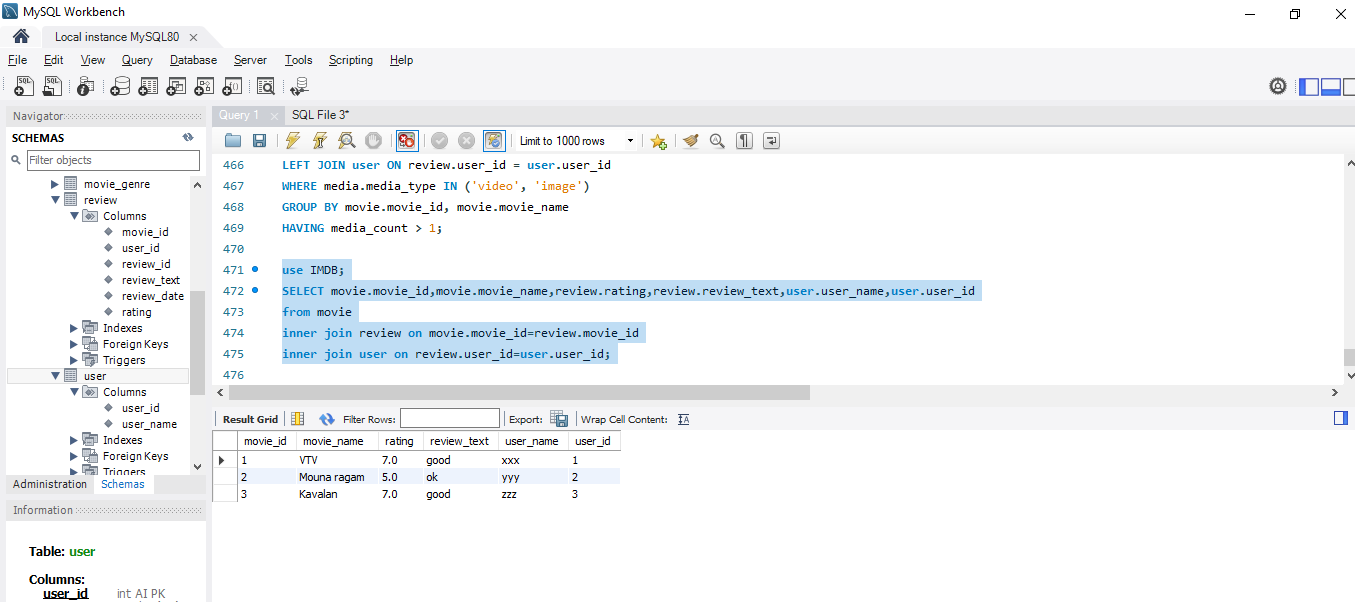
LEFT JOIN review ON movie.movie\_id = review.movie\_id

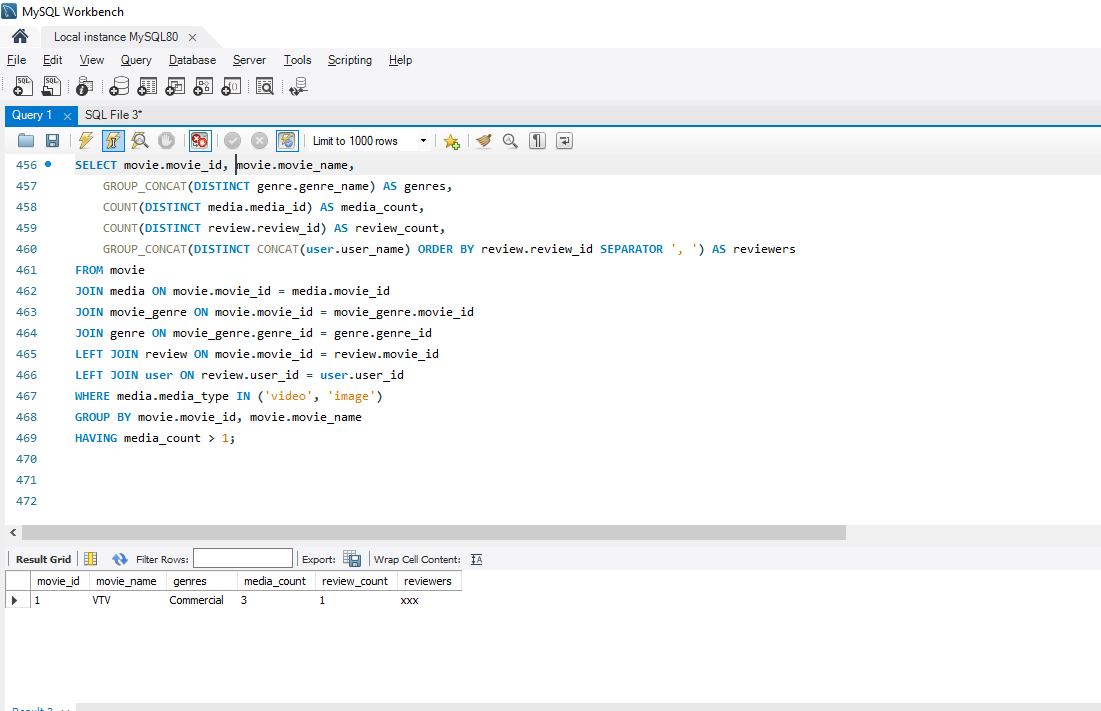
LEFT JOIN user ON review.user\_id = user.user\_id

WHERE media.media\_type IN ('video', 'image')

GROUP BY movie.movie\_id, movie.movie\_name

HAVING media\_count > 1;



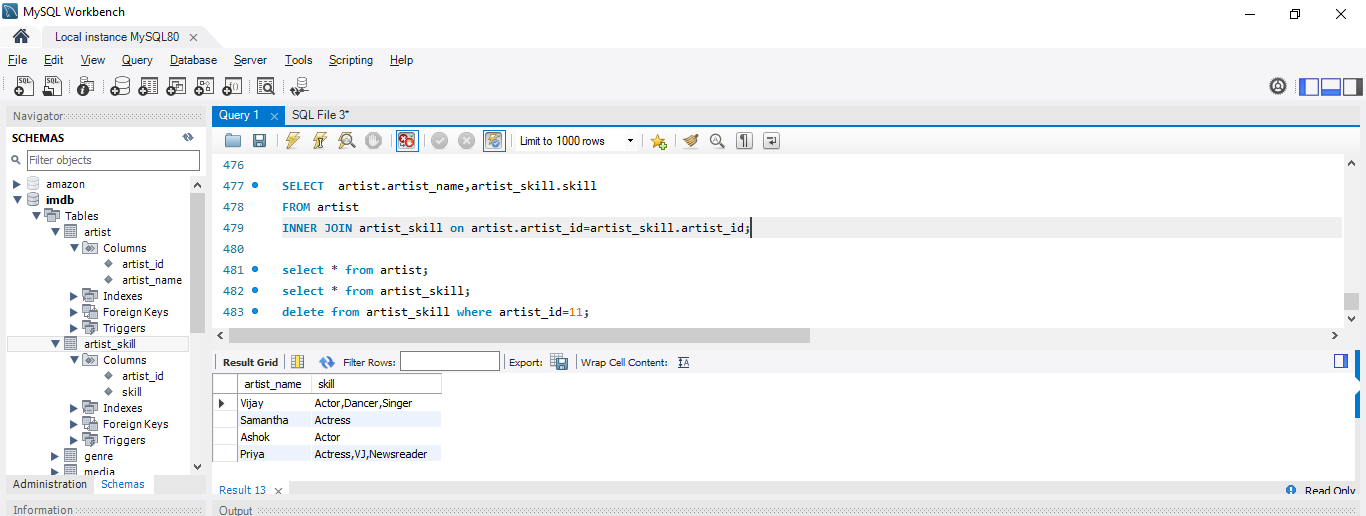


4.Artist can have multiple skills

SELECT artist.artist\_name,artist\_skill.skill

FROM artist

INNER JOIN artist\_skill on artist.artist\_id=artist\_skill.artist\_id;



5.Artist can perform multiple role in a single flim

SELECT movie.movie\_id,movie.movie\_name,movie\_artist.role

from movie

join movie\_artist on movie.movie\_id=movie\_artist.movie\_id;

