#Create Table

CREATE TABLE Movies (

MovieID INT PRIMARY KEY,

Title VARCHAR(100),

Genre VARCHAR(50),

ReleaseYear INT,

Rating DECIMAL(3,1)

);

#Insert Values in Movies Table

INSERT INTO Movies (MovieID, Title, Genre, ReleaseYear, Rating) VALUES

(1, 'The Shawshank Redemption', 'Drama', 1994, 9.3),

(2, 'The Godfather', 'Crime', 1972, 9.2),

(3, 'The Dark Knight', 'Action', 2008, 9.0),

(4, 'Pulp Fiction', 'Crime', 1994, 8.9),

(5, 'Inception', 'Sci-Fi', 2010, 8.8),

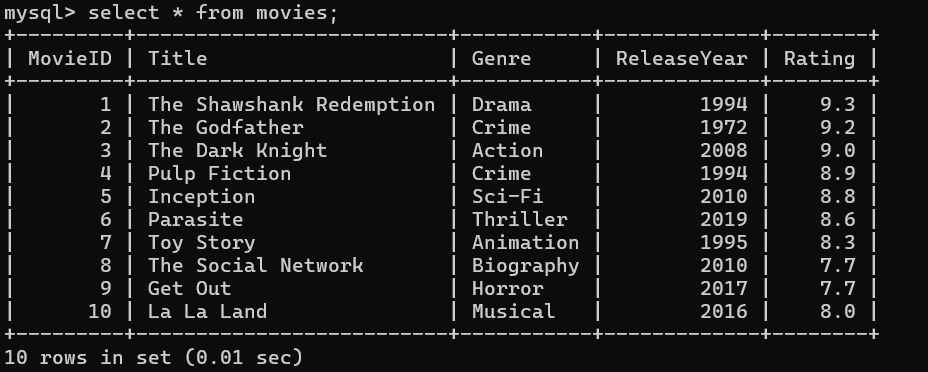
(6, 'Parasite', 'Thriller', 2019, 8.6),

(7, 'Toy Story', 'Animation', 1995, 8.3),

(8, 'The Social Network', 'Biography', 2010, 7.7),

(9, 'Get Out', 'Horror', 2017, 7.7),

(10, 'La La Land', 'Musical', 2016, 8.0);



#Create second table and insert Values in moviedetails

CREATE TABLE MovieDetails (

DetailID INT PRIMARY KEY,

MovieID INT,

Director VARCHAR(100),

Runtime INT, -- in minutes

BoxOffice DECIMAL(12,2), -- in millions

FOREIGN KEY (MovieID) REFERENCES Movies(MovieID)

);

INSERT INTO MovieDetails (DetailID, MovieID, Director, Runtime, BoxOffice) VALUES

(1, 1, 'Frank Darabont', 142, 73.3),

(2, 2, 'Francis Ford Coppola', 175, 291.0),

(3, 3, 'Christopher Nolan', 152, 1006.0),

(4, 4, 'Quentin Tarantino', 154, 213.9),

(5, 5, 'Christopher Nolan', 148, 836.8),

(6, 6, 'Bong Joon Ho', 132, 258.9),

(7, 7, 'John Lasseter', 81, 394.4),

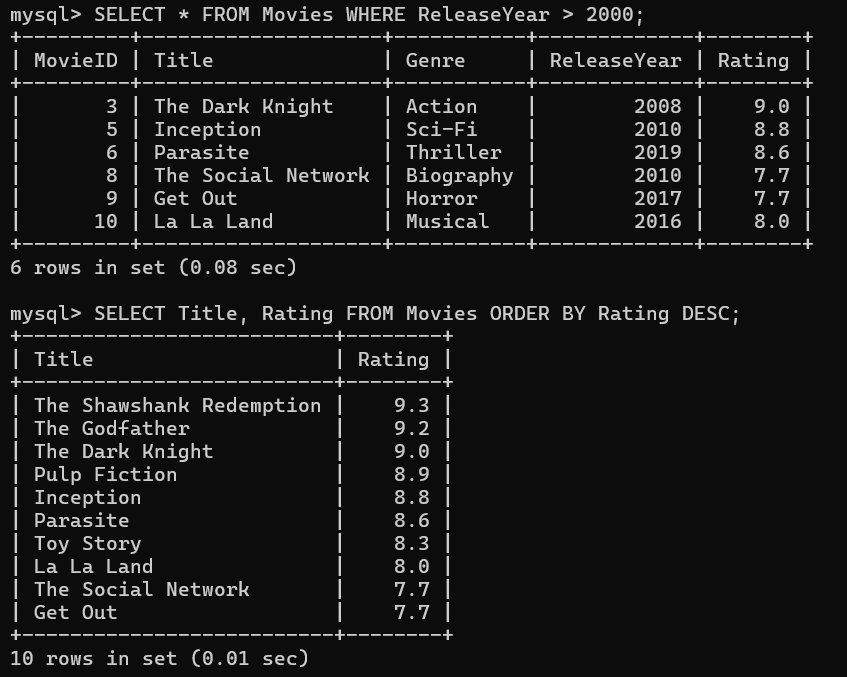
(8, 8, 'David Fincher', 120, 224.9),

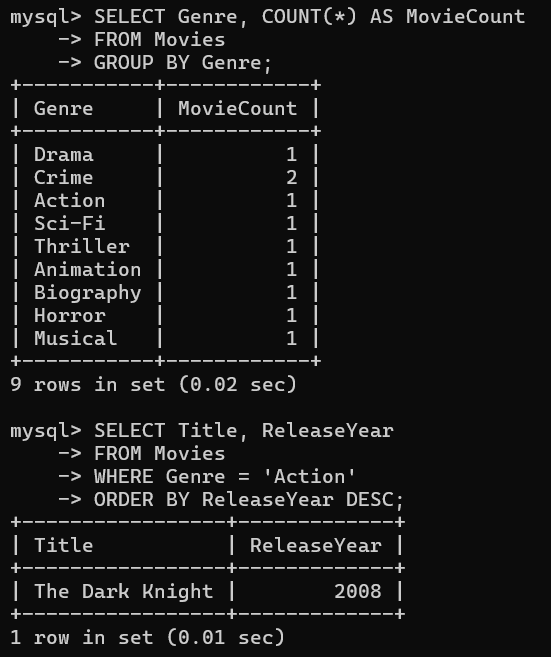
(9, 9, 'Jordan Peele', 104, 255.4),

(10, 10, 'Damien Chazelle', 128, 446.1);

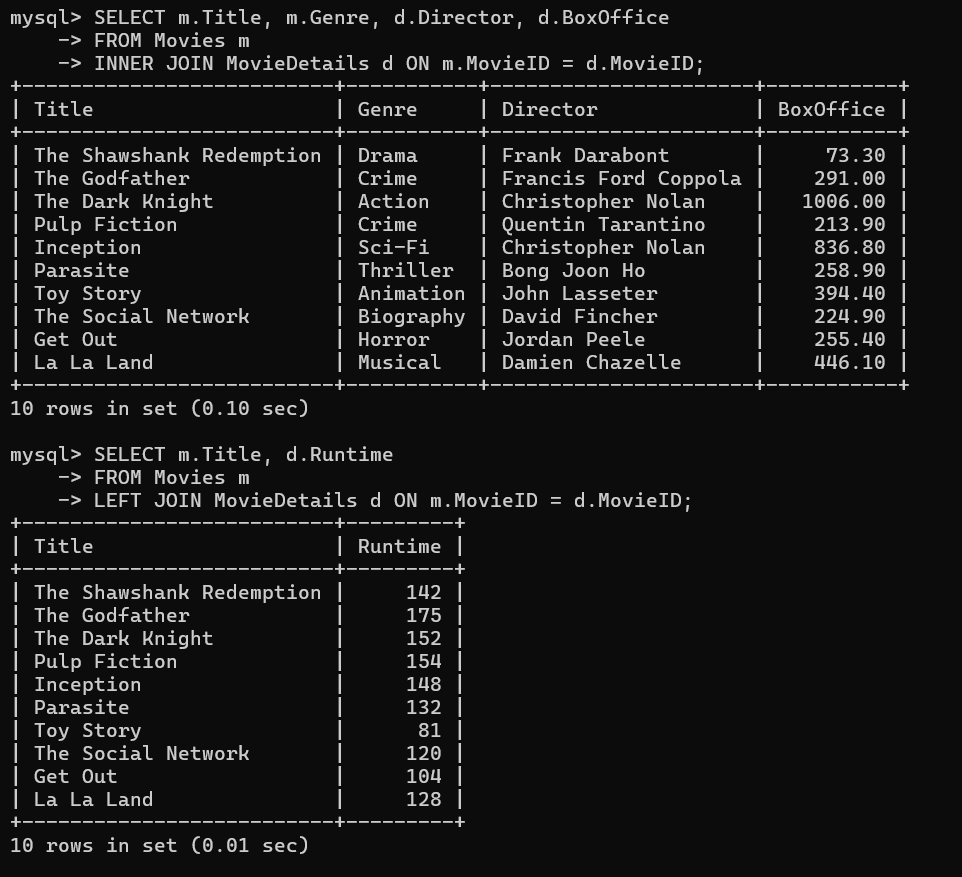


#**Basic SELECT, WHERE, ORDER BY, GROUP BY**



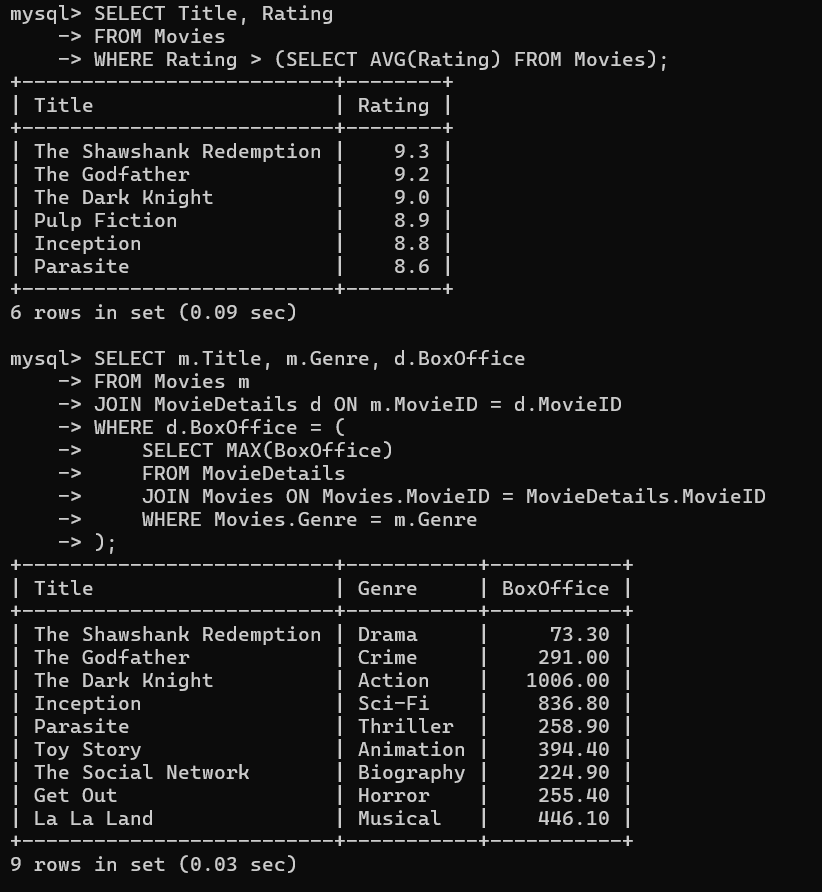


#**b. JOIN Operations (INNER, LEFT, RIGHT)**

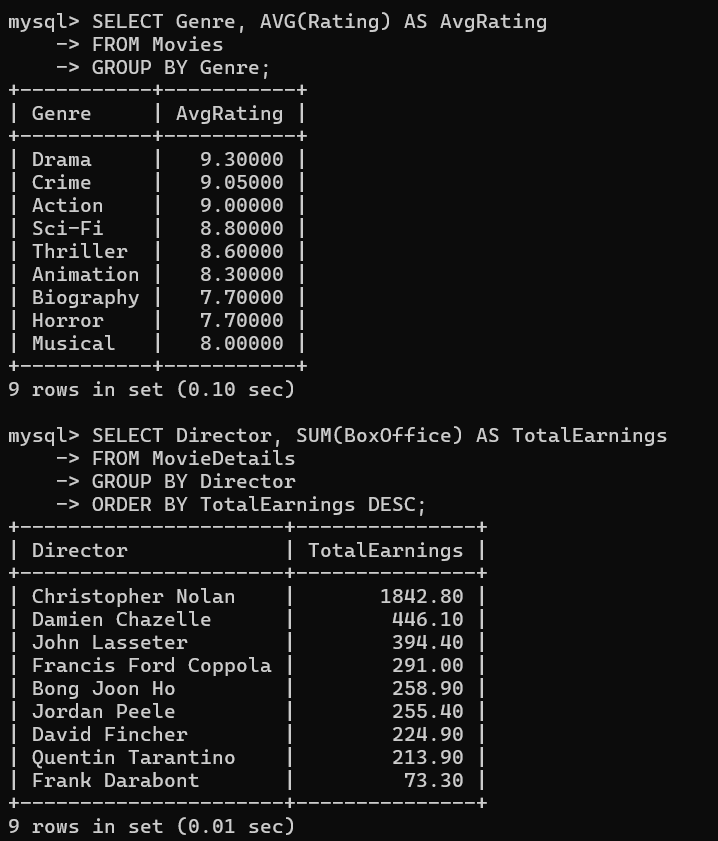


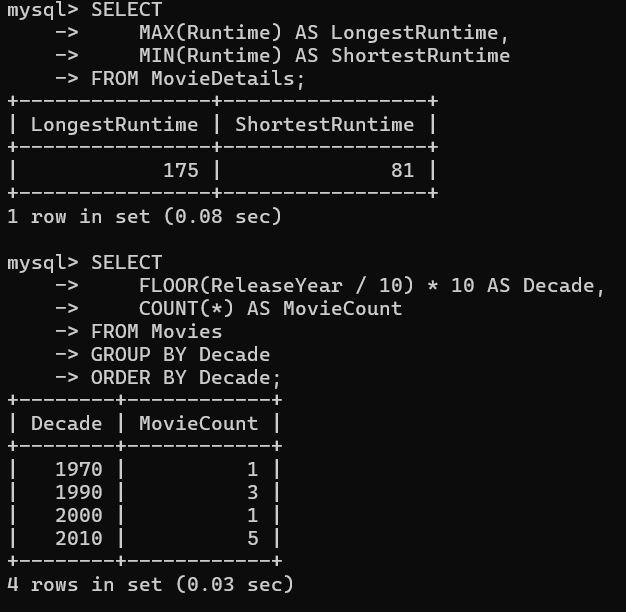


#**c. Subqueries**



**#d. Aggregate Functions (SUM, AVG, COUNT, MAX, MIN)**

****

****

**e. Views for Analysis**

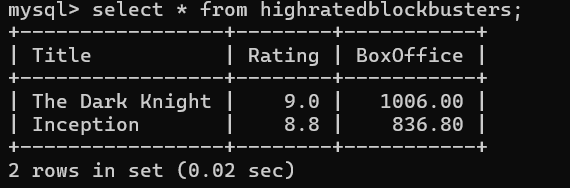
CREATE VIEW HighRatedBlockbusters AS

SELECT m.Title, m.Rating, d.BoxOffice

FROM Movies m

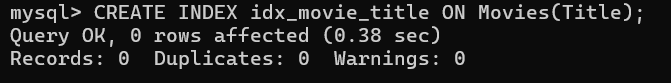
JOIN MovieDetails d ON m.MovieID = d.MovieID

WHERE m.Rating > 8.5 AND d.BoxOffice > 500;



#**f. Query Optimization with Indexes**

-- For WHERE clauses on MovieID (primary key already indexed)



-- For filtering by Genre

