-- Create Actor table

CREATE TABLE Actor (

act\_id INT PRIMARY KEY,

act\_name VARCHAR(50) NOT NULL,

act\_gender CHAR(1) CHECK (act\_gender IN ('M', 'F'))

);

-- Create Director table

CREATE TABLE Director (

did INT PRIMARY KEY,

dname VARCHAR(50) NOT NULL,

d\_phone VARCHAR(15)

);

-- Create Movies table

CREATE TABLE Movies (

mov\_id INT PRIMARY KEY,

mov\_title VARCHAR(100) NOT NULL,

mov\_year INT,

did INT,

FOREIGN KEY (did) REFERENCES Director(did)

);

-- Create Movie\_cast table

CREATE TABLE Movie\_cast (

act\_id INT,

mov\_id INT,

role VARCHAR(50),

PRIMARY KEY (act\_id, mov\_id),

FOREIGN KEY (act\_id) REFERENCES Actor(act\_id),

FOREIGN KEY (mov\_id) REFERENCES Movies(mov\_id)

);

-- Create Ratings table

CREATE TABLE Ratings (

rev\_id INT,

mov\_id INT,

rev\_stars INT CHECK (rev\_stars BETWEEN 1 AND 5),

PRIMARY KEY (rev\_id, mov\_id),

FOREIGN KEY (mov\_id) REFERENCES Movies(mov\_id)

);

-- Insert directors

INSERT INTO Director VALUES

(1, 'Sanjay Leela Bhansali', '9876543210'),

(2, 'Ram Gopal Verma', '8765432109'),

(3, 'Karan Johar', '7654321098'),

(4, 'Zoya Akhtar', '6543210987');

-- Insert actors

INSERT INTO Actor VALUES

(101, 'Ranveer Singh', 'M'),

(102, 'Deepika Padukone', 'F'),

(103, 'Shah Rukh Khan', 'M'),

(104, 'Alia Bhatt', 'F'),

(105, 'Amitabh Bachchan', 'M'),

(106, 'Aishwarya Rai', 'F');

-- Insert movies

INSERT INTO Movies VALUES

(1001, 'Padmaavat', 2018, 1),

(1002, 'Bajirao Mastani', 2015, 1),

(1003, 'Sarkar', 2005, 2),

(1004, 'Dilwale Dulhania Le Jayenge', 1995, 3),

(1005, 'Gully Boy', 2019, 4),

(1006, 'Devdas', 2002, 1);

-- Insert movie cast

INSERT INTO Movie\_cast VALUES

(101, 1001, 'Alauddin Khilji'),

(102, 1001, 'Rani Padmavati'),

(101, 1002, 'Bajirao'),

(102, 1002, 'Mastani'),

(105, 1003, 'Subhash Nagre'),

(101, 1005, 'Murad'),

(104, 1005, 'Safeena'),

(103, 1004, 'Raj'),

(106, 1006, 'Paro'),

(105, 1006, 'Devdas');

-- Insert ratings

INSERT INTO Ratings VALUES

(201, 1001, 4),

(201, 1002, 5),

(202, 1001, 3),

(202, 1003, 2),

(203, 1004, 5),

(203, 1005, 4),

(204, 1006, 5),

(205, 1005, 3);

SELECT m.mov\_title

FROM Movies m

JOIN Director d ON m.did = d.did

WHERE d.dname = 'Sanjay Leela Bhansali';

SELECT DISTINCT a.act\_name

FROM Actor a

JOIN Movie\_cast mc ON a.act\_id = mc.act\_id

JOIN Movies m ON mc.mov\_id = m.mov\_id

WHERE m.mov\_year BETWEEN 2016 AND 2018;

SELECT m.mov\_title, r.rev\_stars

FROM Movies m

JOIN Ratings r ON m.mov\_id = r.mov\_id

ORDER BY r.rev\_stars DESC;

SELECT DISTINCT m.mov\_title

FROM Movies m

JOIN Movie\_cast mc ON m.mov\_id = mc.mov\_id

WHERE mc.act\_id IN (

SELECT act\_id

FROM Movie\_cast

GROUP BY act\_id

HAVING COUNT(mov\_id) >= 2

);

SELECT m.mov\_title, MAX(r.rev\_stars) AS highest\_rating

FROM Movies m

JOIN Ratings r ON m.mov\_id = r.mov\_id

GROUP BY m.mov\_title

ORDER BY m.mov\_title;

UPDATE Ratings

SET rev\_stars = 5

WHERE mov\_id IN (

SELECT m.mov\_id

FROM Movies m

JOIN Director d ON m.did = d.did

WHERE d.dname = 'Ram Gopal Verma'

);

-- Verify the update

SELECT m.mov\_title, r.rev\_stars

FROM Movies m

JOIN Ratings r ON m.mov\_id = r.mov\_id

JOIN Director d ON m.did = d.did

WHERE d.dname = 'Ram Gopal Verma';