**PROGRAM 9: MOVIE DATABASE**

Consider the schema for Movie Database:

**ACTOR**(Act\_id, Act\_Name, Act\_Gender)

**DIRECTOR**(Dir\_id, Dir\_Name, Dir\_Phone)

**MOVIES**(Mov\_id, Mov\_Title, Mov\_Year, Mov\_Lang, Dir\_id) **MOVIE\_CAST**(Act\_id,Mov\_id, Role)

**RATING**(Mov\_id, Rev\_Stars)

Write SQL queries to

1. List the titles of all movies directed by ‘Hitchcock’.

2. Find the movie names where one or more actors acted in two or more movies.

3. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).

4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest

number of stars that movie received. Sort the result by movie title.

5. Update rating of all movies directed by ‘Steven Spielberg’ to 5.

create database m\_movie;

use m\_movie;

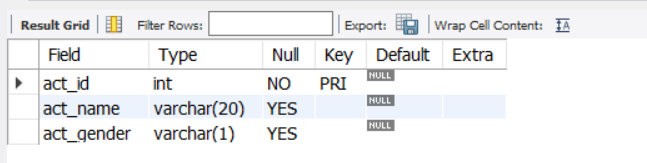
create table actor(act\_id int,

act\_name varchar(20),

act\_gender varchar(1),

primary key(act\_id));

desc actor;



create table director(

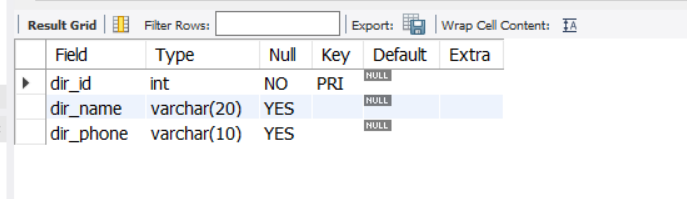
dir\_id int,

dir\_name varchar(20),

dir\_phone varchar(10),

primary key(dir\_id));

desc director;



create table movies(

mov\_id int,

mov\_title varchar(20),

mov\_year varchar(10),

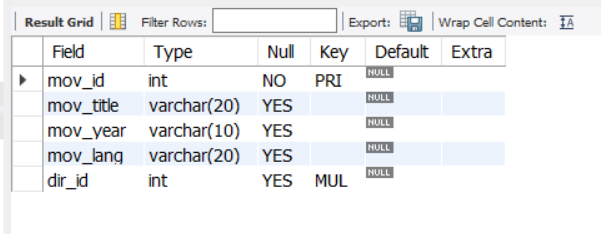
mov\_lang varchar(20),

dir\_id int,

primary key(mov\_id),

foreign key(dir\_id) references director(dir\_id));

desc movies;



create table movie\_cast(

act\_id int,

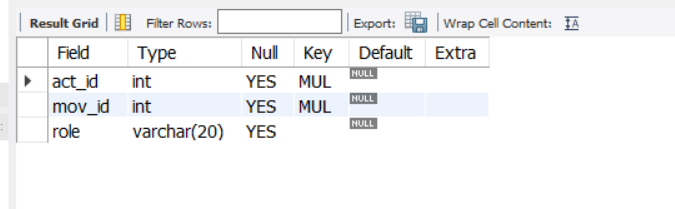
mov\_id int,

role varchar(20),

foreign key(act\_id) references actor(act\_id),

foreign key(mov\_id) references movies(mov\_id));

desc movie\_cast;



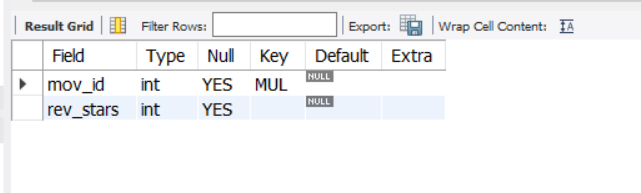
create table rating(

mov\_id int,

rev\_stars int,

foreign key(mov\_id) references movies(mov\_id));

desc rating;



INSERT INTO actor (act\_id, act\_name, act\_gender) VALUES ('101', 'Varun Dhavan', 'M');

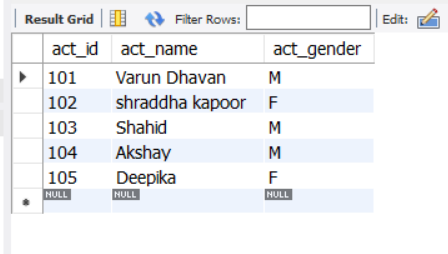
INSERT INTO actor (act\_id, act\_name, act\_gender) VALUES ('102', 'shraddha kapoor', 'F');

INSERT INTO actor (act\_id, act\_name, act\_gender) VALUES ('103', 'Shahid', 'M');

INSERT INTO actor (act\_id, act\_name, act\_gender) VALUES ('104', 'Akshay', 'M');

INSERT INTO actor (act\_id, act\_name, act\_gender) VALUES ('105', 'Deepika', 'F');

select \* from actor;



INSERT INTO director (dir\_id, dir\_name, dir\_phone) VALUES ('11', 'Karan ', '8987565412');

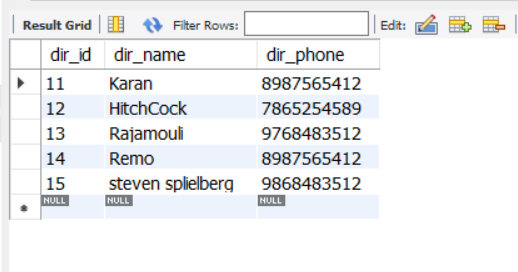
INSERT INTO director (dir\_id, dir\_name, dir\_phone) VALUES ('12', 'HitchCock', '7865254589');

INSERT INTO director (dir\_id, dir\_name, dir\_phone) VALUES ('13', 'Rajamouli ', '9768483512');

INSERT INTO director (dir\_id, dir\_name, dir\_phone) VALUES ('14', 'Remo ', '8987565412');

INSERT INTO director (dir\_id, dir\_name, dir\_phone) VALUES ('15', 'steven splielberg ', '9868483512');

select \* from director;



INSERT INTO movies (mov\_id, mov\_title, mov\_year, mov\_lang, dir\_id) VALUES ('1001', 'Street Dancers', '2020-01-20', 'English', '14');

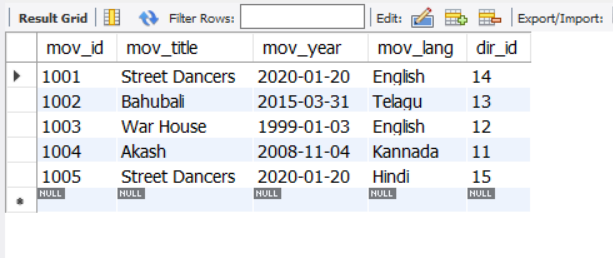
INSERT INTO movies (mov\_id, mov\_title, mov\_year, mov\_lang, dir\_id) VALUES ('1002', 'Bahubali', '2015-03-31', 'Telagu', '13');

INSERT INTO movies (mov\_id, mov\_title, mov\_year, mov\_lang, dir\_id) VALUES ('1003', 'War House', '1999-01-03', 'English', '12');

INSERT INTO movies (mov\_id, mov\_title, mov\_year, mov\_lang, dir\_id) VALUES ('1004', 'Akash', '2008-11-04', 'Kannada', '11');

INSERT INTO movies (mov\_id, mov\_title, mov\_year, mov\_lang, dir\_id) VALUES ('1005', 'Street Dancers', '2020-01-20', 'Hindi', '15');

select \* from movies;



INSERT INTO movie\_cast (act\_id, mov\_id, role) VALUES ('101', '1001', 'Hero');

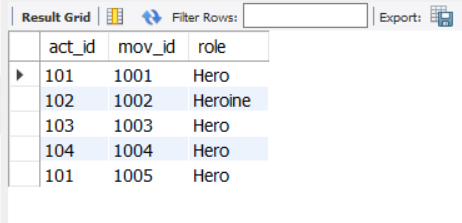
INSERT INTO movie\_cast (act\_id, mov\_id, role) VALUES ('102', '1002', 'Heroine');

INSERT INTO movie\_cast (act\_id, mov\_id, role) VALUES ('103', '1003', 'Hero');

INSERT INTO movie\_cast (act\_id, mov\_id, role) VALUES ('104', '1004', 'Hero');

INSERT INTO movie\_cast (act\_id, mov\_id, role) VALUES ('101', '1005', 'Hero');

select \* from movie\_cast;



INSERT INTO rating (mov\_id, rev\_stars) VALUES ('1001', '5');

INSERT INTO rating (mov\_id, rev\_stars) VALUES ('1002', '1');

INSERT INTO rating (mov\_id, rev\_stars) VALUES ('1003', '4');

INSERT INTO rating (mov\_id, rev\_stars) VALUES ('1004', '2');

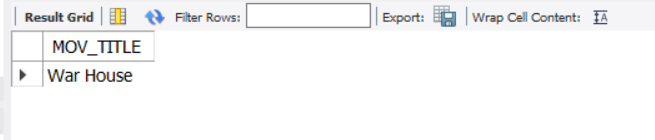
INSERT INTO rating (mov\_id, rev\_stars) VALUES ('1005', '3');

select \* from rating;



-- 1. List the titles of all movies directed by ‘Hitchcock’.

SELECT MOV\_TITLE FROM MOVIES WHERE DIR\_ID IN (SELECT DIR\_ID FROM DIRECTOR WHERE DIR\_NAME = 'HITCHCOCK');



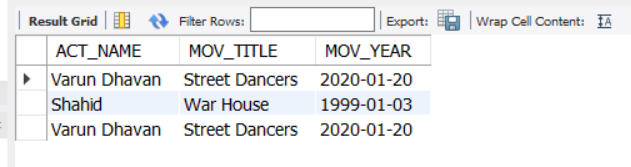
-- 2. Find the movie names where one or more actors acted in two or more movies.

SELECT MOV\_TITLE FROM MOVIES M, MOVIE\_CAST MV WHERE M.MOV\_ID=MV.MOV\_ID AND ACT\_ID IN (SELECT ACT\_ID FROM MOVIE\_CAST GROUP BY ACT\_ID HAVING count(ACT\_ID)>=1) GROUP BY MOV\_TITLE HAVING count(mov\_title)>1;



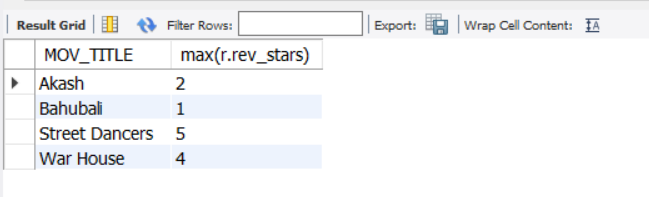
-- 3. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).

SELECT ACT\_NAME, MOV\_TITLE, MOV\_YEAR FROM ACTOR A JOIN MOVIE\_CAST C ON A.ACT\_ID=C.ACT\_ID JOIN MOVIES M ON C.MOV\_ID=M.MOV\_ID WHERE M.MOV\_YEAR NOT BETWEEN 2000 AND 2015;



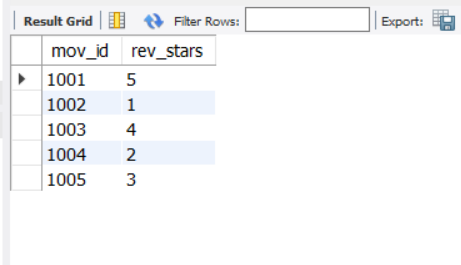
-- 4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.

SELECT m.MOV\_TITLE, max(r.rev\_stars) FROM MOVIES m natural join RATING r where r.rev\_stars>=1 GROUP BY m.MOV\_TITLE ORDER BY m.MOV\_TITLE;



-- 5. Update rating of all movies directed by ‘Steven Spielberg’ to 5

UPDATE RATING SET REV\_STARS=5 WHERE MOV\_ID IN (SELECT MOV\_ID FROM MOVIES WHERE DIR\_ID IN (SELECT DIR\_ID FROM DIRECTOR WHERE DIR\_NAME = 'STEVEN SPIELBERG'));



commit;