**PRATICAL 02**

CREATE TABLE ACTOR (

ACT\_ID INT PRIMARY KEY,

ACT\_NAME VARCHAR (20),

ACT\_GENDER CHAR (1)

);

CREATE TABLE DIRECTOR (

DIR\_ID INT PRIMARY KEY,

DIR\_NAME VARCHAR (20),

DIR\_PHONE INT

);

CREATE TABLE MOVIES (

MOV\_ID INT PRIMARY KEY,

MOV\_TITLE VARCHAR (25),

MOV\_YEAR INT,

MOV\_LANG VARCHAR (12),

DIR\_ID INT,

FOREIGN KEY (DIR\_ID) REFERENCES DIRECTOR (DIR\_ID));

CREATE TABLE MOVIE\_CAST (

ACT\_ID INT,

MOV\_ID INT,

OLE VARCHAR (10),

PRIMARY KEY (ACT\_ID, MOV\_ID),

FOREIGN KEY (ACT\_ID) REFERENCES ACTOR (ACT\_ID),

FOREIGN KEY (MOV\_ID) REFERENCES MOVIES (MOV\_ID));

CREATE TABLE RATING (

MOV\_ID INT,

REV\_STARS VARCHAR (25),

PRIMARY KEY (MOV\_ID),

FOREIGN KEY (MOV\_ID) REFERENCES MOVIES (MOV\_ID));

INSERT INTO ACTOR VALUES (301,'ANUSHKA','F');

INSERT INTO ACTOR VALUES (302,'PRABHAS','M');

INSERT INTO ACTOR VALUES (303,'PUNITH','M');

INSERT INTO ACTOR VALUES (304,'JERMY','M');

INSERT INTO DIRECTOR VALUES (60,'RAJAMOULI', 87516110); INSERT INTO DIRECTOR VALUES (61,'HITCHCOCK', 77661389);

INSERT INTO DIRECTOR VALUES (62,'FARAN', 99867765);

INSERT INTO DIRECTOR VALUES (63,'STEVEN SPIELBERG', 89897765);

INSERT INTO MOVIES VALUES (1001,'BAHUBALI-2', 2017, 'TELAGU', 60);

INSERT INTO MOVIES VALUES (1002,'BAHUBALI-1', 2015, 'TELAGU', 60); INSERT INTO MOVIES VALUES (1003,'AKASH', 2008, 'KANNADA', 61);

INSERT INTO MOVIES VALUES (1004,'WAR HORSE', 2011, 'ENGLISH', 63);

INSERT INTO MOVIE\_CAST VALUES (301, 1002, 'HEROINE');

INSERT INTO MOVIE\_CAST VALUES (301, 1001, 'HEROINE');

INSERT INTO MOVIE\_CAST VALUES (303, 1003, 'HERO');

INSERT INTO MOVIE\_CAST VALUES (303, 1002, 'GUEST');

INSERT INTO MOVIE\_CAST VALUES (304, 1004, 'HERO');

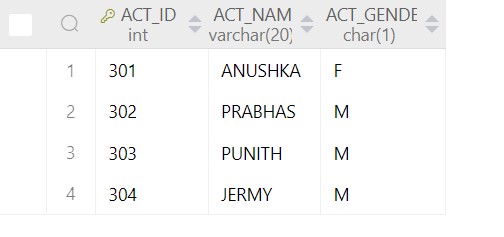
INSERT INTO RATING VALUES (1001, 4);

INSERT INTO RATING VALUES (1002, 2);

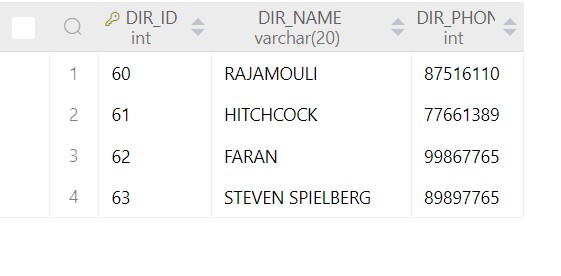
INSERT INTO RATING VALUES (1003, 5);

INSERT INTO RATING VALUES (1004, 4);

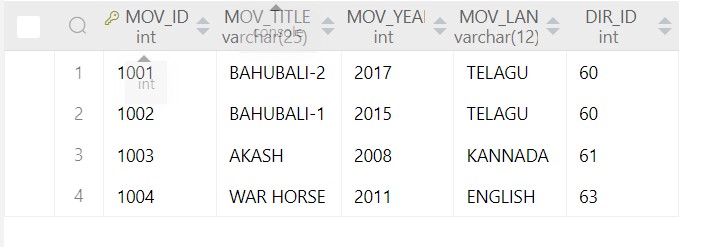
SELECT \* FROM actor;



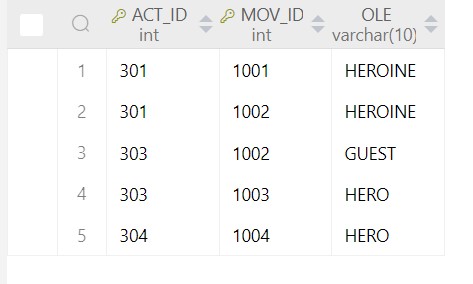
SELECT \* FROM director;



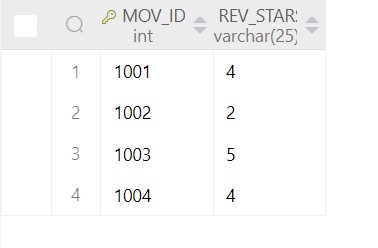
SELECT \* FROM MOVIES;



SELECT \* FROM MOVIE\_CAST;



SELECT \* FROM RATING;



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

SELECT MOV\_TITLE

FROM MOVIES

WHERE DIR\_ID = (SELECT DIR\_ID

FROM DIRECTOR

WHERE DIR\_NAME='HITCHCOCK');



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

SELECT MOV\_TITLE

FROM MOVIES M,MOVIE\_CAST MC

WHERE M.MOV\_ID=MC.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(\*)>1;



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

SELECT ACT\_NAME

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;



**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 MOV\_TITLE,MAX(REV\_STARS)

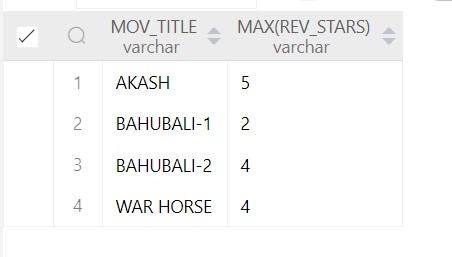
FROM MOVIES

INNER JOIN RATING USING (MOV\_ID)

GROUP BY MOV\_TITLE

HAVING MAX(REV\_STARS)>0

ORDER BY MOV\_TITLE;



**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'));

Q3)

CREATE TABLE students ( stno INT PRIMARY KEY, name VARCHAR(50), addr VARCHAR(255), city VARCHAR(50), state VARCHAR(2), zip VARCHAR(10)

);

CREATE TABLE INSTRUCTORS ( empno INT PRIMARY KEY, names VARCHAR(50), ranks VARCHAR(20), roomno VARCHAR(10), telno VARCHAR(15)

);

CREATE TABLE COURSES ( cno INT PRIMARY KEY, cname VARCHAR(50), cr INT, cap INT

);

CREATE TABLE GRADES ( stno INT, empno INT, cno INT, sem VARCHAR(10), year INT, grade INT,

PRIMARY KEY (stno),

FOREIGN KEY (stno) REFERENCES students(stno),

FOREIGN KEY (empno) REFERENCES INSTRUCTORS(empno),

FOREIGN KEY (cno) REFERENCES COURSES(cno)

);

CREATE TABLE ADVISING ( stno INT, empno INT,

PRIMARY KEY (stno, empno),

FOREIGN KEY (stno) REFERENCES students(stno),

FOREIGN KEY (empno) REFERENCES INSTRUCTORS(empno)

);

-- Insert records into the COURSES table

INSERT INTO COURSES (cno, cname, cr, cap)

VALUES

(1, 'Math101', 3, 30),

(2, 'CS210', 4, 25),

(3, 'Physics101', 3, 20);

-- Insert records into the GRADES table

INSERT INTO GRADES (stno, empno, cno, sem, year, grade)

VALUES

(1, 101, 1, 'Fall', 2021, 85),

(2, 102, 2, 'Fall', 2021, 92),

(3, 103, 3, 'Fall', 2021, 78);

-- Insert records into the ADVISING table

INSERT INTO ADVISING (stno, empno)

VALUES

(1, 101),

(2, 102),

(3, 103);