

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)

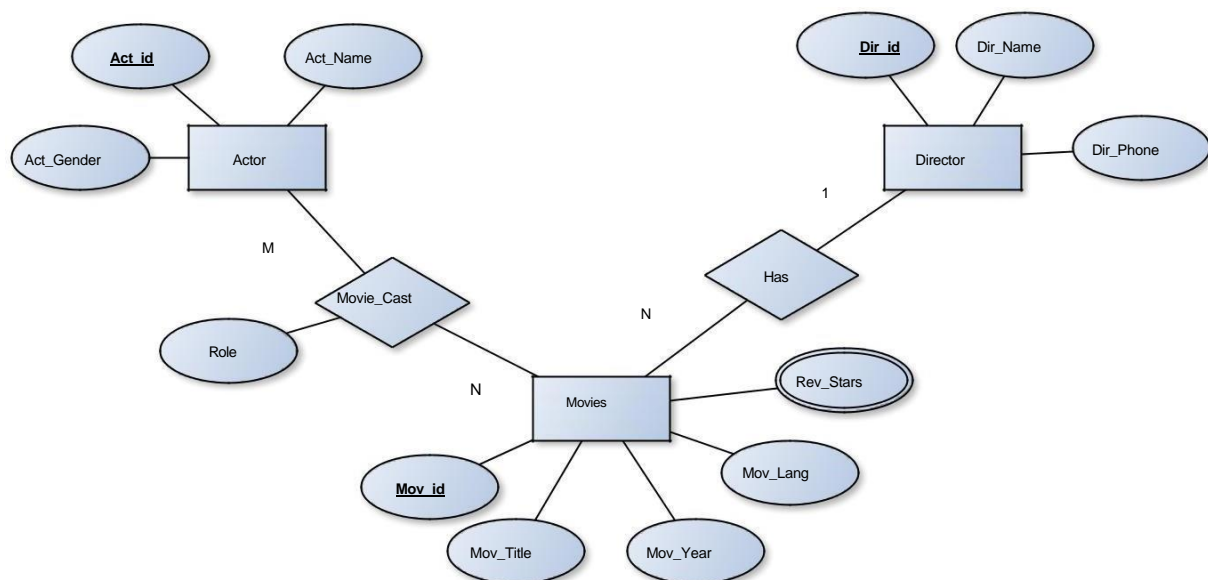
RATING (Mov_id, Rev_Stars)

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.

Solution:

Entity-Relationship Diagram



Schema Diagram

Actor

<u>Act_id</u>	Act_Name	Act_Gender
---------------	----------	------------

Director

<u>Dir_id</u>	Dir_Name	Dir_Phone
---------------	----------	-----------

Movies

<u>Mov_id</u>	Mov_Title	Mov_Year	Mov_Lang	Dir_id
---------------	-----------	----------	----------	--------

Movie_Cast

<u>Act_id</u>	<u>Mov_id</u>	Role
---------------	---------------	------

Rating

<u>Mov_id</u>	Rev_Stars
---------------	-----------

Create Table ACTOR with Primary Key as ACT_ID

```
CREATE TABLE ACTOR (  
  ACT_ID INTEGER PRIMARY KEY,  
  ACT_NAME VARCHAR(20),  
  ACT_GENDER CHAR(1));
```

DESC ACTOR;

Create Table DIRECTOR with Primary Key as DIR_ID

```
CREATE TABLE DIRECTOR(  
  DIR_ID INTEGER PRIMARY KEY,  
  DIR_NAME VARCHAR(20),  
  DIR_PHONE INTEGER);
```

DESC DIRECTOR;

Create Table MOVIES with Primary Key as MOV_ID and Foreign Key DIR_ID referring DIRECTOR table

```
CREATE TABLE MOVIES(  
MOV_ID INTEGER PRIMARY KEY,  
MOV_TITLE VARCHAR(25),  
MOV_YEAR INTEGER,  
MOV_LANG VARCHAR(15),  
DIR_ID INTEGER,  
FOREIGN KEY (DIR_ID) REFERENCES DIRECTOR(DIR_ID));  
  
DESC MOVIES;
```

Create Table MOVIE_CAST with Primary Key as MOV_ID and ACT_ID and Foreign Key ACT_ID and MOV_ID referring ACTOR and MOVIES tables respectively

```
CREATE TABLE MOVIE_CAST(  
ACT_ID INTEGER,  
MOV_ID INTEGER,  
ROLE VARCHAR(10),  
PRIMARY KEY (ACT_ID,MOV_ID),  
FOREIGN KEY (ACT_ID) REFERENCES ACTOR(ACT_ID),  
FOREIGN KEY (MOV_ID) REFERENCES MOVIES(MOV_ID));  
  
DESC MOVIE_CAST;
```

Create Table RATING with Primary Key as MOV_ID and Foreign Key MOV_ID referring MOVIES table

```
CREATE TABLE RATING(  
MOV_ID INTEGER PRIMARY KEY,  
REV_STARS VARCHAR(25),  
FOREIGN KEY (MOV_ID) REFERENCES MOVIES(MOV_ID));
```

```
DESC RATING;
```

INSERTION OF VALUES:

Inserting records into ACTOR table

```
INSERT INTO ACTOR VALUES(101,'RAHUL','M');  
INSERT INTO ACTOR VALUES(102,'ANKITHA','F');  
INSERT INTO ACTOR VALUES(103,'RADHIKA','F');  
INSERT INTO ACTOR VALUES(104,'CHETHAN','M');  
INSERT INTO ACTOR VALUES(105,'VIVAN','M');
```

```
SELECT * FROM ACTOR;
```

Inserting records into DIRECTOR table

```
INSERT INTO DIRECTOR VALUES(201,'ANUP',918181818);  
INSERT INTO DIRECTOR VALUES(202,'HITCHCOCK',918181812);  
INSERT INTO DIRECTOR VALUES(203,'SHASHANK',918181813);  
INSERT INTO DIRECTOR VALUES(204,'STEVEN SPIELBERG',918181814);  
INSERT INTO DIRECTOR VALUES(205,'ANAND',918181815);
```

```
SELECT * FROM DIRECTOR;
```

Inserting records into MOVIES table

```
INSERT INTO MOVIES VALUES(1001,'MANASU',2017,'KANNADA',201);  
INSERT INTO MOVIES VALUES(1002,'AAKASHAM',2015,'TELUGU',202);  
INSERT INTO MOVIES VALUES(1003,'KALIYONA',2008,'KANNADA',201);  
INSERT INTO MOVIES VALUES(1004,'WAR HORSE',2011,'ENGLISH',204);  
INSERT INTO MOVIES VALUES(1005,'HOME',2012,'ENGLISH',205);
```

```
SELECT * FROM MOVIES;
```

Inserting records into MOVIE_CAST table

```
INSERT INTO MOVIE_CAST VALUES(101,1002,'HERO');  
INSERT INTO MOVIE_CAST VALUES(101,1001,'HERO');  
INSERT INTO MOVIE_CAST VALUES(103,1003,'HEROINE');  
INSERT INTO MOVIE_CAST VALUES(103,1002,'GUEST');  
INSERT INTO MOVIE_CAST VALUES(104,1004,'HERO');
```

```
SELECT * FROM MOVIE_CAST;
```

Inserting records into RATING table

```
INSERT INTO RATING VALUES(1001,4);  
INSERT INTO RATING VALUES(1002,2);  
INSERT INTO RATING VALUES(1003,5);  
INSERT INTO RATING VALUES(1004,4);  
INSERT INTO RATING VALUES(1005,3);
```

```
SELECT * FROM RATING;
```

1)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');
```

2)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;
```

3)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;
```

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 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;
```

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'));
```