

**A. 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)

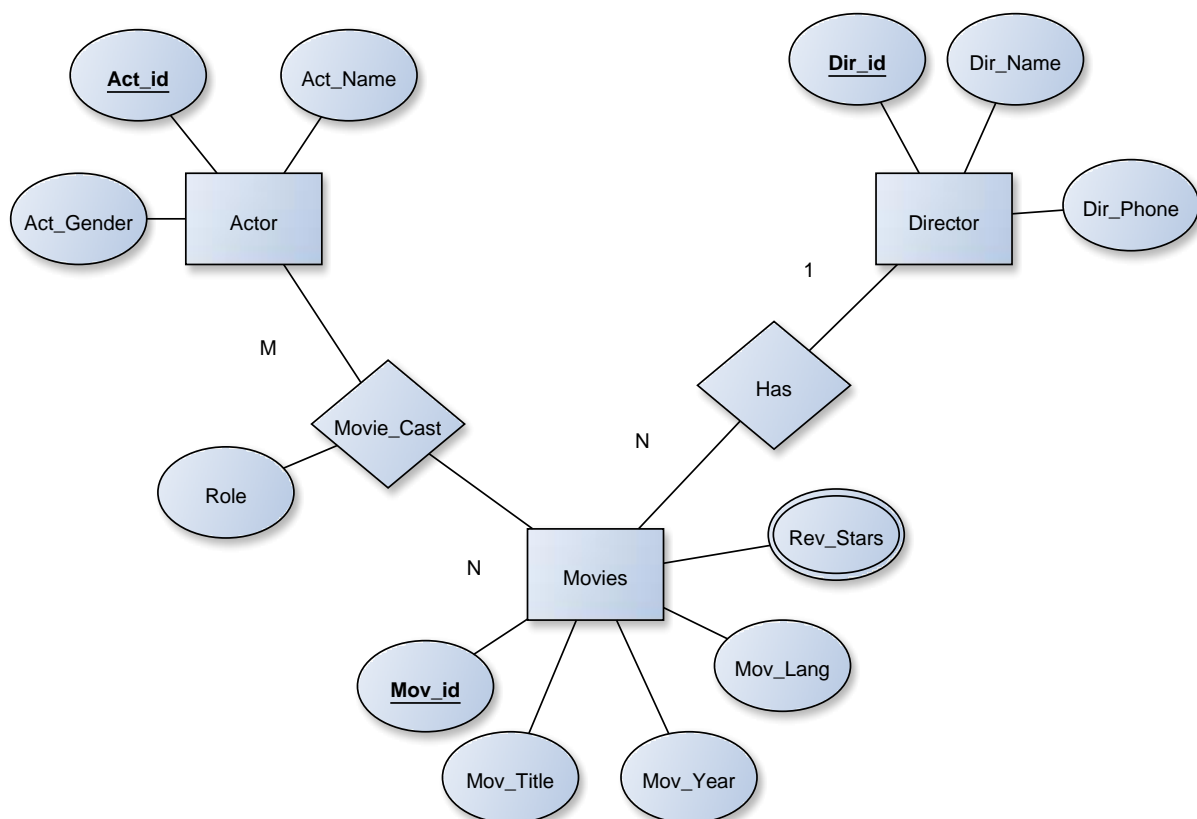
**RATING** (Mov\_id, Rev\_Stars)

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
---------------	-----------

## Table Creation

```
CREATE TABLE ACTOR (  
  ACT_ID INTEGER (3),  
  ACT_NAME    VARCHAR (20),  
  ACT_GENDER CHAR (1),  
  PRIMARY KEY (ACT_ID));
```

```
CREATE TABLE DIRECTOR (  
  DIR_ID INTEGER (3),  
  DIR_NAME VARCHAR (20),  
  DIR_PHONE    INTEGER (10),  
  PRIMARY KEY (DIR_ID));
```

```
CREATE TABLE MOVIES (  
  MOV_ID INTEGER (4),  
  MOV_TITLE    VARCHAR (25),  
  MOV_YEAR     INTEGER (4),  
  MOV_LANG     VARCHAR (12),  
  DIR_ID INTEGER (3),  
  PRIMARY KEY (MOV_ID),  
  FOREIGN KEY (DIR_ID) REFERENCES DIRECTOR (DIR_ID));
```

```
CREATE TABLE MOVIE_CAST (  
  ACT_ID INTEGER (3),  
  MOV_ID INTEGER (4),  
  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));
```

```
CREATE TABLE RATING (  
  MOV_ID INTEGER (4),  
  REV_STARS    INTEGER (1),  
  PRIMARY KEY (MOV_ID));
```

MOV\_ID INTEGER (4),  
 REV\_STARS VARCHAR (25),  
 FOREIGN KEY (MOV\_ID) REFERENCES MOVIES (MOV\_ID));

### Table Descriptions

DESC ACTOR;

SQL> DESC ACTOR;

Name	Null?	Type
ACT_ID	NOT NULL	NUMBER(3)
ACT_NAME		VARCHAR2(20)
ACT_GENDER		CHAR(1)

DESC DIRECTOR;

SQL> DESC DIRECTOR;

Name	Null?	Type
DIR_ID	NOT NULL	NUMBER(3)
DIR_NAME		VARCHAR2(20)
DIR_PHONE		NUMBER(10)

DESC MOVIES;

SQL> DESC MOVIES;

Name	Null?	Type
MOV_ID	NOT NULL	NUMBER(4)
MOV_TITLE		VARCHAR2(25)
MOV_YEAR		NUMBER(4)
MOV_LANG		VARCHAR2(12)
DIR_ID		NUMBER(3)

DESC MOVIE\_CAST;

SQL> DESC MOVIE\_CAST;

Name	Null?	Type
ACT_ID	NOT NULL	NUMBER(3)
MOV_ID	NOT NULL	NUMBER(4)
ROLE		VARCHAR2(10)

DESC RATING;

SQL> DESC RATING;

Name	Null?	Type
MOV_ID	NOT NULL	NUMBER(4)
REV_STARS		VARCHAR2(25)

### Insertion of Values to Tables

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', 8751611001);
INSERT INTO DIRECTOR VALUES (61,'HITCHCOCK', 7766138911);
INSERT INTO DIRECTOR VALUES (62,'FARAN', 9986776531);
INSERT INTO DIRECTOR VALUES (63,'STEVEN SPIELBERG', 8989776530);
```

```
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);
```

```
SQL> SELECT * FROM ACTOR;
```

ACT_ID	ACT_NAME	A
301	ANUSHKA	F
302	PRABHAS	M
303	PUNITH	M
304	JERMY	M

```
SQL> SELECT * FROM DIRECTOR;
```

DIR_ID	DIR_NAME	DIR_PHONE
60	RAJAMOULI	8751611001
61	HITCHCOCK	7766138911
62	FARAN	9986776531
63	STEVEN SPIELBERG	8989776530

```
SQL> SELECT * FROM MOVIES;
```

MOV_ID	MOV_TITLE	MOV_YEAR	MOV_LANG	DIR_ID
1001	BAHUBALI-2	2017	TELAGU	60
1002	BAHUBALI-1	2015	TELAGU	60
1003	AKASH	2008	KANNADA	61
1004	WAR HORSE	2011	ENGLISH	63

```
SQL> SELECT * FROM MOVIE_CAST;
```

ACT_ID	MOV_ID	ROLE
301	1002	HEROINE
301	1001	HEROINE
303	1003	HERO
303	1002	GUEST
304	1004	HERO

```
SQL> SELECT * FROM RATING;
```

MOV_ID	REV_STARS
1001	4
1002	2
1003	5
1004	4

### Queries:

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');
```

MOV_TITLE
AKASH

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 (*)>1;
```

MOV_TITLE
BAHUBALI-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;
```

OR

```

SELECT A.ACT_NAME, A.ACT_NAME, C.MOV_TITLE, C.MOV_YEAR
FROM ACTOR A, MOVIE_CAST B, MOVIES C
WHERE A.ACT_ID=B.ACT_ID
AND B.MOV_ID=C.MOV_ID
AND C.MOV_YEAR NOT BETWEEN 2000 AND 2015;

```

ACT_NAME	MOV_TITLE	MOV_YEAR
ANUSHKA	BAHUBALI-2	2017

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;

```

MOV_TITLE	MAX(REV_STARS)
AKASH	5
BAHUBALI-1	2
BAHUBALI-2	4
WAR HORSE	4

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'));

```

```
SQL> SELECT * FROM RATING;
```

MOV_ID	REV_STARS
1001	4
1002	2
1003	5
1004	5

6. Delete all the tuples whose review stars is less than 3.

```
DELETE FROM RATING WHERE REV_STARS < 3;
```