

Lab Exercise 2

1. 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_CASTING (Act_id, Mov_id, Role)
RATING (Mov_id, Rev_Stars)

Write SQL queries to

1. List the titles of all movies directed by 'Joss Whedon'.
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 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 'Jon Favreau' to 5 star.

Aim:

To create a movie database and execute the SQL queries.

Procedure:

1. Go to database Home page.
2. Create a table with necessary entities.
3. Insert values in the existing table.
4. Execute the selection, list and update query on the appropriate table.
5. Exit.

Entity-Relationship Diagram:

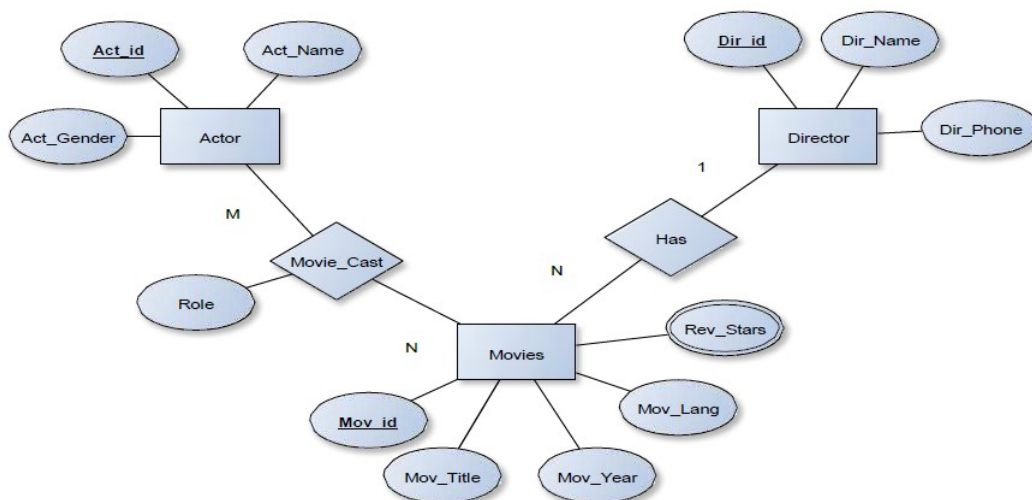


Table Creation:

```
create database movies;
use movies;
CREATE TABLE ACTOR (ACT_ID INT, ACT_NAME VARCHAR (20),
ACT_GENDER CHAR, PRIMARY KEY (ACT_ID));

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

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

CREATE TABLE MOVIE_CAST (ACT_ID INT, MOV_ID INT,
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 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', 8751611001);
INSERT INTO DIRECTOR VALUES (61,'JOSS WHEDON', 7766138911);
INSERT INTO DIRECTOR VALUES (62,'FARAN', 9986776531);
INSERT INTO DIRECTOR VALUES (63,'JON FAVREAU', 8989776530);
```

```

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

```

OUTPUT:

Query 1

List the titles of all movies directed by 'Joss Whedon'.

```

SELECT MOV_TITLE FROM MOVIES WHERE DIR_ID IN (SELECT DIR_ID
FROM DIRECTOR WHERE DIR_NAME ='Joss Whedon')

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
MOV_TITLE			
AKASH			

Query 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;

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
MOV_TITLE			
BAHUBALI-1			

Query 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;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
ACT_NAME	MOV_TITLE	MOV_YEAR	
ANUSHKA	BAHUBALI-2	2017	

Query 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;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
MOV_TITLE	MAX(REV_STARS)		
AKASH	5		
BAHUBALI-1	2		
BAHUBALI-2	4		
WAR HORSE	4		

Query 5:

Update rating of all movies directed by 'Jon Favreau' 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 = 'Jon Favreau'));
```

52 12:26:33 UPDATE RATING SET REV_STARS=5 WHERE MOV_ID IN (SELECT MOV_ID FROM M... 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 [Activate W](#)

Result:

Thus the library databases with SQL queries were executed successfully.