

## DBMS LAB 6

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 Movie_db;  
USE Movie_db;
```

```
CREATE TABLE Actor(  
    act_id int PRIMARY KEY,  
    act_name VARCHAR(50),  
    act_gender VARCHAR(50)  
);
```

```
CREATE TABLE Director(  
    dir_id int PRIMARY KEY,  
    dir_name VARCHAR(50),  
    dir_phone VARCHAR(50)  
);
```

```
CREATE TABLE Movies(  
    mov_id int PRIMARY KEY,  
    mov_title VARCHAR(50),  
    mov_year year,  
    mov_lang VARCHAR(50),  
    dir_id int,  
    FOREIGN KEY (dir_id) REFERENCES Director(dir_id)
```

```
);
CREATE TABLE Movie_cast (
    act_id int,
    mov_id int,
    role VARCHAR(50),
    FOREIGN KEY (act_id) REFERENCES Actor(act_id),
    FOREIGN KEY (mov_id) REFERENCES Movies(mov_id),
    PRIMARY KEY (act_id,mov_id)
);
CREATE TABLE Rating(
    mov_id int PRIMARY KEY ,
    rev_stars int,
    FOREIGN KEY (mov_id) REFERENCES Movies(mov_id)
);
```

```
INSERT INTO Actor VALUES(1001, 'Tom Crusie','M'),(1002, 'Jason Statham','M'),(1003, 'Maia Mitchell','F'),(1004, 'Margot Robbie','F'),(1005, 'kristine stewart','F'),(1006, 'Ryan Reynolds','M');
INSERT INTO Director VALUES(9001, 'Hitchcock',9874562154),(9002, 'Martin Scorsese',9874560054),(9003, 'Christopher Nolan',9874562178),(9004, 'Christopher Loyd',9874564454),(9005, 'James Cameron',9874562994),(9006, 'Woody Allen',9874503154);
INSERT INTO Movies
VALUES(101,'Passengers',2014,'English',9001),(102,'Dark',2001,'German',9001),(103,'Spiderman',1998,'English',9002),(104,'Lucifer',1999,'English',9003),(105,'After We Collided',2017,'English',9002),(106,'Captain America',1994,'English',9004);
INSERT INTO Movie_cast VALUES
(1001,101,'Jack'),(1001,102,'Hardin'),(1002,102,'Elon'),(1003,103,'Tessa'),(1004,104,'Claire'),(1006,105,'Martha'),(1005,106,'Charlie'),(1002,106,'Craft'),(1002,104,'Michael'),(1005,105,'Aurora');
INSERT INTO Rating VALUES(101,4),(102,3),(103,5),(104,2),(105,4),(106,3);
```

-- QUERY1            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');
```

-- QUERY2            Find the movie names WHERE one or more actors acted in two or more movies.

```
SELECT distinct m.mov_title,c.act_id FROM Movies m, Movie_cast c WHERE
m.mov_id=c.mov_id and c.act_id in (SELECT act_id FROM Movie_cast GROUP BY act_id
having count(mov_id)>1);
```

-- QUERY3            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 WHERE act_id in(SELECT a.act_id FROM (SELECT act_id
FROM Movie_cast NATURAL JOIN Movies WHERE mov_year<2000)a inner join (SELECT
act_id FROM Movie_cast NATURAL JOIN Movies WHERE mov_year>2015)b on
a.act_id=b.act_id);
```

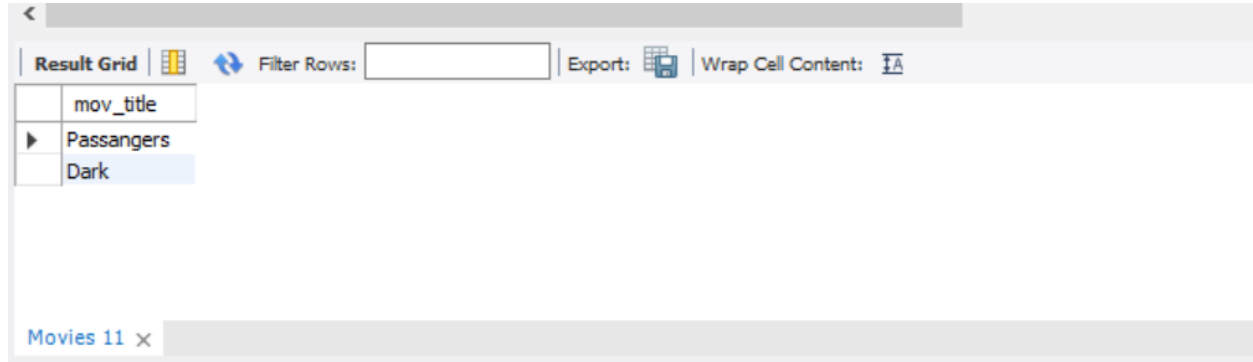
-- QUERY4 Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.

```
SELECT mov_title,max(rev_stars) FROM Movies m,Rating r WHERE m.mov_id=r.mov_id
GROUP BY m.mov_title having count(r.rev_stars)>0;
```

-- QUERY5 Update rating of all movies directed by 'Martin Scorsese' to 5.

```
UPDATE Rating SET rev_stars=5 WHERE mov_id in (SELECT mov_id FROM Movies inner join
Director on Movies.dir_id=Director.dir_id and Director.dir_name='Martin Scorsese');
SELECT * FROM Rating;
```

## QUERY 1:







mov_title	Passangers	Dark

Movies 11 x

## QUERY 2:

<

Result Grid   Filter Rows:  | Export:  | Wrap Cell Content: 

	mov_title	act_id
▶	Passangers	1001
	Dark	1001
	Dark	1002
	Lucifer	1002
	After We Collided	1005
	Captain America	1002
	Captain America	1005

Result 12 x

### QUERY 3:

<

Result Grid   Filter Rows:  | Export:  | Wrap Cell Content: 





	act_name
▶	kristine stewart

Actor 13 x

Output

### QUERY 4:

<




Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content: 

	mov_title	max(rev_stars)
▶	Passangers	4
	Dark	3
	Spiderman	5
	Lucifer	2
	After We Collided	5
	Captain America	3

Result 14 ×

### QUERY 5:

<

Result Grid |   Filter Rows:  | Edit: 

	mov_id	rev_stars
▶	101	4
	102	3
	103	5
	104	2
	105	5
	106	3
✱	NULL	NULL

Rating 15 ×

Output