

DBMS LAB Program -3

3 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)

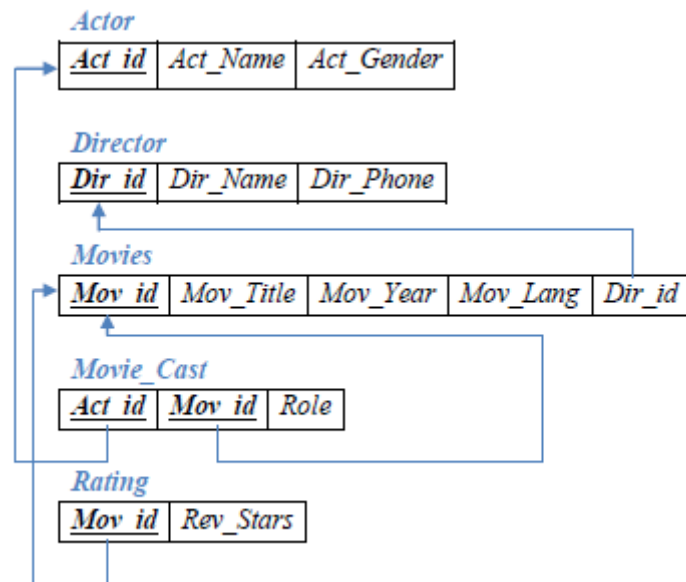
rating (mov_id, rev_stars)

Write SQL queries to

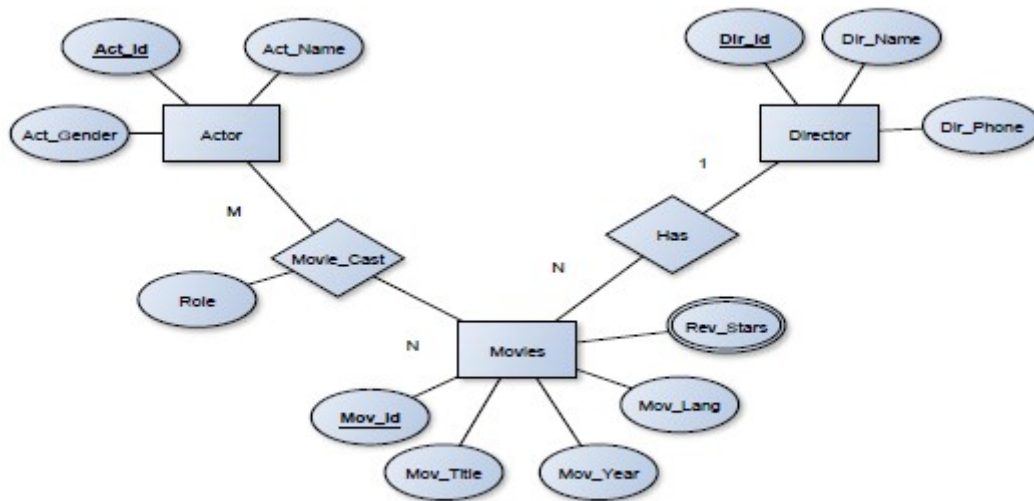
1. List the titles of all movies directed by 'Hitchcock'.
2. Find the movie names where one or more actors acted in two or more movies.
3. List all actors who acted in a movie before 2000 and also in a movie after 2015
(use JOIN operation).
4. Find the title of movies and number of stars for each movie 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:

Schema Diagram:



Entity – Relationship Diagram



Creating Tables (Relations):

```
mysql>create database moviedb;
```

```
mysql>use moviedb;
```

actor (act_id, act_name, act_gender)

- create table **actor**(act_id int primary key, act_name varchar(20), act_gender char(1));

```
mysql> desc actor;
```

Field	Type	Null	Key	Default	Extra
act_id	int(11)	NO	PRI	NULL	
act_name	varchar(20)	YES		NULL	
act_gender	char(1)	YES		NULL	

director(dir_id, dir_name, dir_phone)

- create table **director**(dir_id int primary key, dir_name varchar(20), dir_phone int);

```
mysql> desc director;
```

Field	Type	Null	Key	Default	Extra
dir_id	int(11)	NO	PRI	NULL	
dir_name	varchar(20)	YES		NULL	
dir_phone	int(11)	YES		NULL	

- create table **movies**(mov_id int primary key, mov_title varchar(25), mov_year year, mov_lang varchar(12), dir_id int, foreign key (dir_id) references director(dir_id));

mysql> desc movies;

Field	Type	Null	Key	Default	Extra
mov_id	int(11)	NO	PRI	NULL	
mov_title	varchar(25)	YES		NULL	
mov_year	year(4)	YES		NULL	
mov_lang	varchar(12)	YES		NULL	
dir_id	int(11)	YES	MUL	NULL	

- 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));

mysql> desc movie_cast;

Field	Type	Null	Key	Default	Extra
act_id	int(11)	NO	PRI	NULL	
mov_id	int(11)	NO	PRI	NULL	
role	varchar(10)	YES		NULL	

- create table **rating**(mov_id int primary key, rev_stars varchar(25), foreign key(mov_id) references movies(mov_id));

mysql> desc rating;

Field	Type	Null	Key	Default	Extra
mov_id	int(11)	NO	PRI	NULL	
rev_stars	varchar(25)	YES		NULL	

Insertion of Values into the Tables

actor Table

```
mysql>insert into actor values (301,'anushka','f');
```

```
mysql>insert into actor values (302,'prabhas','m');
```

```
mysql>insert into actor values (303,'punith','m');
```

```
mysql>insert into actor values (304,'jermy','m');
```

```
mysql> select * from actor;
```

act_id	act_name	act_gender
301	anushka	f
302	prabhas	m
303	punith	m
304	jermy	m

director Table

```
mysql>insert into director values (60,'rajamouli', 1251611001);
```

```
mysql>insert into director values (61,'hitchcock', 1166138911);
```

```
mysql>insert into director values (62,'faran', 1246776531);
```

```
mysql>insert into director values (63,'steven spielberg', 2129776530);
```

```
mysql> select * from director;
```

dir_id	dir_name	dir_phone
60	rajamouli	1251611001
61	hitchcock	1166138911
62	faran	1246776531
63	steven spielberg	2129776530

movies Table

```
mysql>insert into movies values (1001,'bahubali-2', 2017, 'telagu', 60);
```

```
mysql>insert into movies values (1002,'bahubali-1', 2015, 'telagu', 60);
```

```
mysql>insert into movies values (1003,'akash', 2008, 'kannada', 61);
```

```
mysql>insert into movies values (1004,'war horse', 2011, 'english', 63);
```

```
mysql> 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

movie_cast Table

```
mysql>insert into movie_cast values (301, 1002, 'heroine');
```

```
mysql>insert into movie_cast values (301, 1001, 'heroine');
```

```
mysql>insert into movie_cast values (303, 1003, 'hero');
```

```
mysql>insert into movie_cast values (303, 1002, 'guest');
```

```
mysql>insert into movie_cast values (304, 1004, 'hero');
```

```
mysql> select * from movie_cast;
```

act_id	mov_id	role
301	1001	heroine
301	1002	heroine
303	1002	guest
303	1003	hero
304	1004	hero

rating Table

```
mysql>insert into rating values (1001, 4);
```

```
mysql>insert into rating values (1002, 2);
```

```
mysql>insert into rating values (1003, 5);
```

```
mysql>insert into rating values (1004, 4);
```

```
mysql> 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 m.mov_title
```

```
from movies m, director d
```

```
where d.dir_id = m.dir_id and dir_name = 'hitchcock';
```

mov_title
akash

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

```
select m.mov_title
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 (act_id) > 1)
group by m.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 a.act_name, m.mov_title, m.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;
```

```
+-----+-----+-----+
| 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 m.mov_title, max(r.rev_stars)
from movies m, rating r
where m.mov_id = r.mov_id
group by m.mov_title
having max(r.rev_stars) > 0
order by mov_title;
```

mov_title	max(r.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 m.mov_id from movies m, director d
where m.dir_id = d.dir_id and d.dir_name = 'steven spielberg');
```

Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from rating;

mov_id	rev_stars
1001	4
1002	2
1003	5
1004	5