Lab Program 3

Consider the schema for Movie Database:

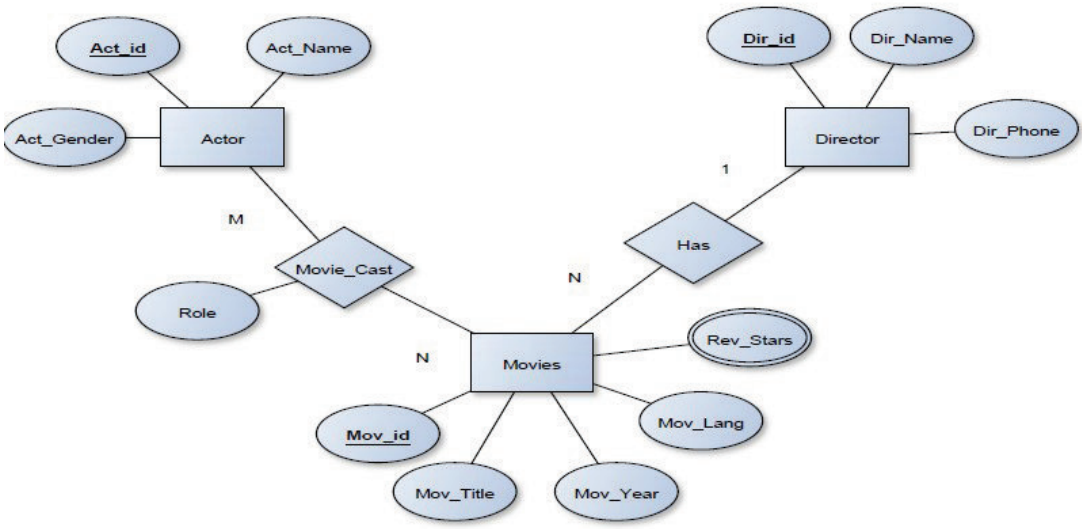
ACTOR(Act_id, Act_Name, Act_Gender)
DIRECTOR(Dir_id, Dir_Name, Dir_Phone)
MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)
MOVIE_CAST(Act_id, Mov_id, Role)
RATING(Mov_id, Rev_Stars)

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.

AIM: Create table, querying the Movie database and perform all the operations using sql.

ER DIAGRAM:



SCHEMA DIAGRAM:

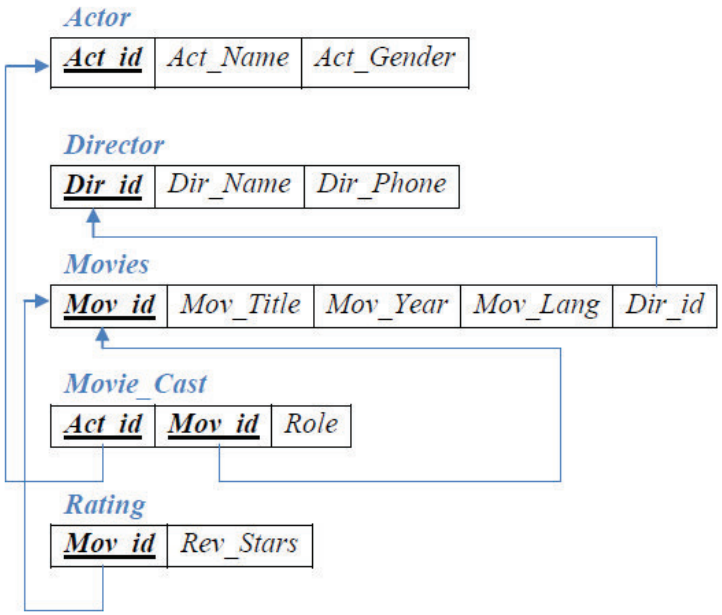


TABLE CREATION

mysql> create table actor(act_id int(5) primary key,act_name varchar(20),act_gender varchar(5));
Query OK, 0 rows affected (0.09 sec)

mysql> create table director(director_id int(5) primary key,director_name varchar(20),director_phone double);
Query OK, 0 rows affected (0.09 sec)

mysql> create table movies(mov_id int(5) primary key,mov_title varchar(20),mov_year int,mov_language varchar(20),director_id int(5),foreign key(director_id) references director(director_id) on delete cascade);
Query OK, 0 rows affected (0.08 sec)

mysql> create table movie_cast(act_id int(5),mov_id int(5),role varchar(20),primary key(act_id,mov_id),foreign key(act_id) references actor(act_id) on delete cascade,foreign key(mov_id) references movies(mov_id) on delete cascade);
Query OK, 0 rows affected (0.10 sec)

mysql> create table rating(mov_id int(5) primary key,rev_stars decimal(2,1),foreign key(mov_id) references movies(mov_id) on delete cascade);
Query OK, 0 rows affected (0.07 sec)

mysql> insert into actor values(01,'rajesh','M');
Query OK, 1 row affected (0.06 sec)

mysql> insert into actor values(02,'akshay','M');
Query OK, 1 row affected (0.04 sec)

mysql> insert into actor values(03,'aishwarya','F');
Query OK, 1 row affected (0.04 sec)

```
mysql> insert into actor values(04,'deepika','F');
Query OK, 1 row affected (0.05 sec)

mysql> insert into director values(100,'rakesh',89526358536);
Query OK, 1 row affected (0.04 sec)

mysql> insert into director values(101,'karan',8952635567);
Query OK, 1 row affected (0.04 sec)

mysql> insert into director values(103,'aktar',8952635577);
Query OK, 1 row affected (0.05 sec)

mysql> insert into director values(104,'bansali',9952635577);
Query OK, 1 row affected (0.04 sec)

mysql> insert into movies values(200,'knph',2000,'hindi',100);
Query OK, 1 row affected (0.04 sec)

mysql> insert into movies values(201,'k3g',2001,'hindi',101);
Query OK, 1 row affected (0.04 sec)

mysql> insert into movies values(203,'kirik party',2016,'kannada',103);
Query OK, 1 row affected (0.05 sec)

mysql> insert into movies values(204,'khamoshi',1994,'hindi',104);
Query OK, 1 row affected (0.04 sec)

mysql> insert into movie_cast values(01,200,'hero');
Query OK, 1 row affected (0.03 sec)

mysql> insert into movie_cast values(02,201,'hero');
Query OK, 1 row affected (0.04 sec)

mysql> insert into movie_cast values(03,203,'heroine');
Query OK, 1 row affected (0.04 sec)

mysql> insert into movie_cast values(04,204,'heroine');
Query OK, 1 row affected (0.04 sec)

mysql>  insert into movie_cast values(03,201,'heroine');
Query OK, 1 row affected (0.04 sec)

mysql>  insert into movie_cast values(01,204,'hero');
Query OK, 1 row affected (0.04 sec)

mysql> insert into rating values(200,3.5);
Query OK, 1 row affected (0.05 sec)

mysql> insert into rating values(201,1.5);
Query OK, 1 row affected (0.03 sec)

mysql> insert into rating values(203,4);
Query OK, 1 row affected (0.05 sec)

mysql> insert into rating values(204,4.5);
```

Query OK, 1 row affected (0.04 sec)

```
mysql> select * from actor;
+--------+-----------+------------+
| act_id | act_name  | act_gender |
+--------+-----------+------------+
|      1 | rajesh    | M          |
|      2 | akshay    | M          |
|      3 | aishwarya | F          |
|      4 | deepika   | F          |
+--------+-----------+------------+
4 rows in set (0.00 sec)

mysql> select * from director;
+-------------+---------------+----------------+
| director_id | director_name | director_phone |
+-------------+---------------+----------------+
|         100 | rakesh        |    8952635853 |
|         101 | karan         |    8952635567 |
|         103 | aktar         |    8952635577 |
|         104 | bansali       |    9952635577 |
+-------------+---------------+----------------+
4 rows in set (0.00 sec)

mysql> select * from movies;
+--------+-------------+----------+--------------+-------------+
| mov_id | mov_title   | mov_year | mov_language | director_id |
+--------+-------------+----------+--------------+-------------+
|    200 | knph        |     2000 | hindi        |         100 |
|    201 | k3g         |     2001 | hindi        |         101 |
|    203 | kirik party |     2016 | kannada      |         103 |
|    204 | khamoshi    |     1994 | hindi        |         104 |
+--------+-------------+----------+--------------+-------------+
4 rows in set (0.00 sec)


mysql> select * from movie_cast;
+--------+--------+---------+
| act_id | mov_id | role    |
+--------+--------+---------+
|      1 |    200 | hero    |
|      1 |    204 | hero    |
|      2 |    201 | hero    |
|      3 |    201 | heroine |
|      3 |    203 | heroine |
|      4 |    204 | heroine |
+--------+--------+---------+
6 rows in set (0.01 sec)

mysql> select * from rating;
+--------+-----------+
| mov_id | rev_stars |
+--------+-----------+
|    200 |       3.5 |
|    201 |       1.5 |
```

```
|   203 |      4.0 |
|   204 |      4.5 |
+--------+-----------+
4 rows in set (0.00 sec)
```

QUERY 1: List the titles of all movies directed by ‘Hitchcock’.

```
mysql> select mov_title from movies m,director d where m.director_id=d.director_id and
d.director_name ='bansali';
+-----------+
| mov_title |
+-----------+
| khamoshi  |
+-----------+
1 row in set (0.00 sec)
```

QUERY 2:

```
mysql> 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);
+-------------+--------+
| mov_title   | act_id |
+-------------+--------+
| knph        |      1 |
| k3g         |      3 |
| kirik party |      3 |
| khamoshi    |      1 |
+-------------+--------+
4 rows in set (0.00 sec)
```

QUERY 3: Find the movie names where one or more actors acted in two or more movies.

```
mysql> 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;
+-----------+-------------+----------+
| act_name  | mov_title   | mov_year |
+-----------+-------------+----------+
| aishwarya | kirik party |     2016 |
| rajesh    | khamoshi    |     1994 |
| deepika   | khamoshi    |     1994 |
+-----------+-------------+----------+
3 rows in set (0.00 sec)
```

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.

```
mysql> select m.mov_title,rev_stars from movies m,rating r where m.mov_id=r.mov_id and
r.rev_stars>0 order by mov_title;
+-------------+-----------+
| mov_title   | rev_stars |
+-------------+-----------+
| k3g         |       1.5 |
| k3g         |       1.5 |
| khamoshi    |       4.5 |
```

```
| khamoshi    |       4.5 |
| kirik party |       4.0 |
| knph        |       3.5 |
+-------------+-----------+
6 rows in set (0.02 sec)
```

QUERY 5: Update rating of all movies directed by ‘Steven Spielberg’ to 5.

```
mysql> update rating set rev_stars= 5 where mov_id in(select mov_id from movies m ,director d
where d.director_id=m.director_id and director_name='rakesh');
Query OK, 1 row affected (0.09 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from rating;
+--------+-----------+
| mov_id | rev_stars |
+--------+-----------+
|    200 |       5.0 |
|    201 |       1.5 |
|    203 |       4.0 |
|    204 |       4.5 |
+--------+-----------+
4 rows in set (0.00 sec)
```

PS: Insert a tuple with director name as Steven Spielberg

CONCLUSION: Tables are created and the values have been inserted accordingly and all the mentioned queries have been executed.