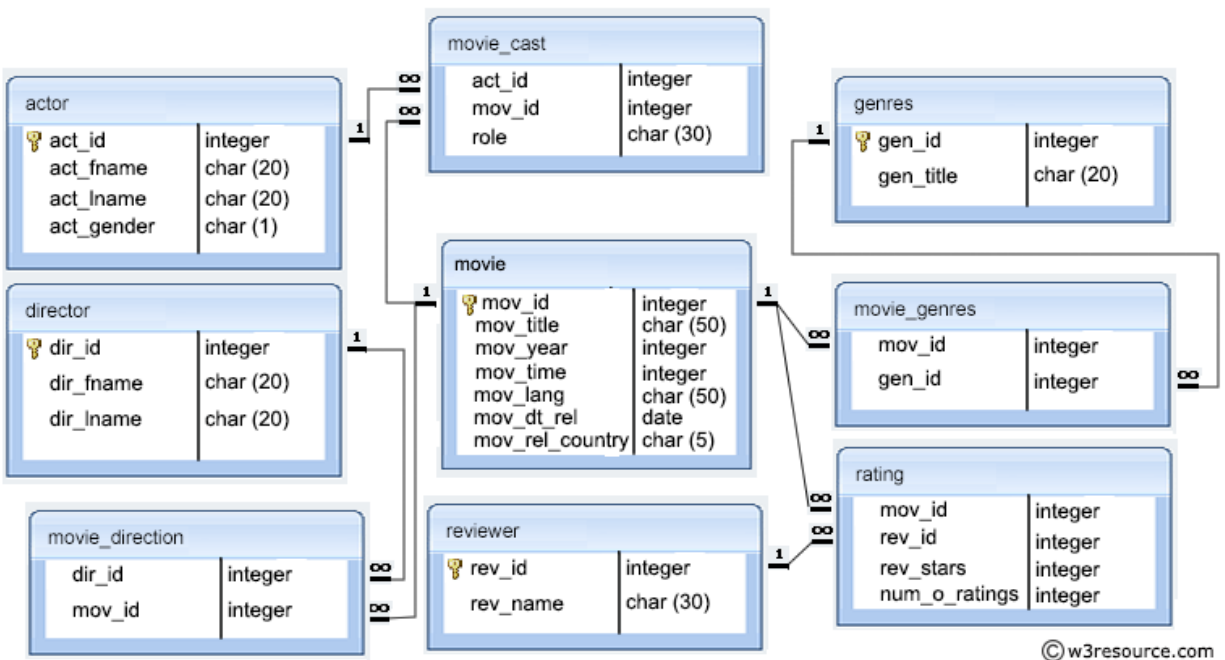


Day-12

W3RESOURCES SQL

MOVIES DATABASE



Structure of Movie Database

Database can be downloaded from W3 Resources and also from my Github account

<https://github.com/im-amit-kumar/100-DAYS-OF-DATA-SCIENCE/tree/main/Day-12>

1. From the following tables, write a SQL query to find the name of all reviewers who have rated their ratings with a NULL value. Return reviewer name.

```
select rev_name
from reviewer r
join rating r1
on r.rev_id= r1.rev_id
where rev_stars is NULL;
```

- 2. From the following tables, write a SQL query to find the actors who were cast in the movie 'Annie Hall'. Return actor first name, last name and role.**

```
select act_fname , act_lname , role
from actor a join movie_cast m
on a.act_id = m.act_id
join movie m1
on m.mov_id = m1.mov_id
where mov_title like '%Annie Hall%';
```

- 3. From the following tables, write a SQL query to find the director who directed a movie that casted a role for 'Eyes Wide Shut'. Return director first name, last name and movie title.**

```
select d.dir_fname , d.dir_lname , m1.mov_title
from director d
join movie_direction md
on d.dir_id = md.dir_id
```

```
join movie_cast mc
on mc.mov_id = md.mov_id
join movie m1
on mc.mov_id = m1.mov_id
where role is not null
and m1.mov_title ='Eyes Wide Shut';
```

4. **From the following tables, write a SQL query to find who directed a movie that casted a role as 'Sean Maguire'. Return director first name, last name and movie title.**

```
select d.dir_fname , d.dir_lname , m1.mov_title
from director d
join movie_direction md
on d.dir_id = md.dir_id
join movie_cast mc
on mc.mov_id = md.mov_id
join movie m1
on mc.mov_id = m1.mov_id
where role is not null
and mc.role ='Sean Maguire';
```

5. **From the following tables, write a SQL query to find the actors who have not acted in any movie between 1990 and 2000 (Begin and end values are included.). Return actor first name, last name, movie title and release year.**

```
select act_fname , act_lname , mov_title , mov_year
```

```
from actor a
join movie_cast mc
on mc.act_id = a.act_id
join movie m
on m.mov_id = mc.mov_id
where mov_year not between 1990 and 2000
order by mov_year;
```

6. **From the following tables, write a SQL query to find the directors with number of genres movies. Group the result set on director first name, last name and generic title. Sort the result-set in ascending order by director first name and last name. Return director first name, last name and number of genres movies.**

```
select d.dir_fname , d.dir_lname , g.gen_title , count(g.gen_title)
from director d
join movie_direction md
on d.dir_id = md.dir_id
join movie_genres mg
on mg.mov_id = md.mov_id
join genres g
on mg.gen_id = g.gen_id
group by dir_fname , dir_lname , gen_title
order by dir_fname , dir_lname;
```

7. **From the following tables, write a SQL query to find the movies with year and genres. Return movie title, movie year and generic title.**

```
select mov_title , mov_year , gen_title  
  
from movie  
  
natural join movie_genres  
  
natural join genres;
```

8. **From the following tables, write a SQL query to find all the movies with year, genres, and name of the director.**

```
select mov_title , mov_year, gen_title , dir_fname , dir_lname  
  
from movie  
  
natural join movie_genres  
  
natural join genres  
  
natural join movie_direction  
  
natural join director;
```

9. **From the following tables, write a SQL query to find the movies released before 1st January 1989. Sort the result-set in descending order by date of release. Return movie title, release year, date of release, duration, and first and last name of the director.**

```
select mov_title , mov_year , mov_dt_rel , mov_time, dir_fname,dir_lname  
from movie m  
join movie_direction md  
    on m.mov_id = md.mov_id  
join director d
```

```
on d.dir_id = md.dir_id
where mov_dt_rel < '01/01/1989'
order by mov_dt_rel desc;
```

- 10. From the following table, write a SQL query to compute the average time and count number of movies for each genre. Return genre title, average time and number of movies for each genre.**

```
select gen_title , avg(mov_time) , count(gen_title)
from genres g
join movie_genres mg
on mg.gen_id = g.gen_id
join movie m
on m.mov_id = mg.mov_id
group by gen_title ;
```

- 11. From the following table, write a SQL query to find movies with the lowest duration. Return movie title, movie year, director first name, last name, actor first name, last name and role.**

```
select mov_title, mov_year, dir_fname, dir_lname,
act_fname, act_lname, role
from movie
natural join movie_direction
natural join movie_cast
natural join director
natural join actor
where mov_time=(SELECT MIN(mov_time) FROM movie);
```

12. From the following table, write a SQL query to find those years when a movie received a rating of 3 or 4. Sort the result in increasing order on movie year. Return move year.

```
select distinct mov_year
from movie
join rating
using (mov_id)
where rev_stars in (3,4)
order by mov_year;
```