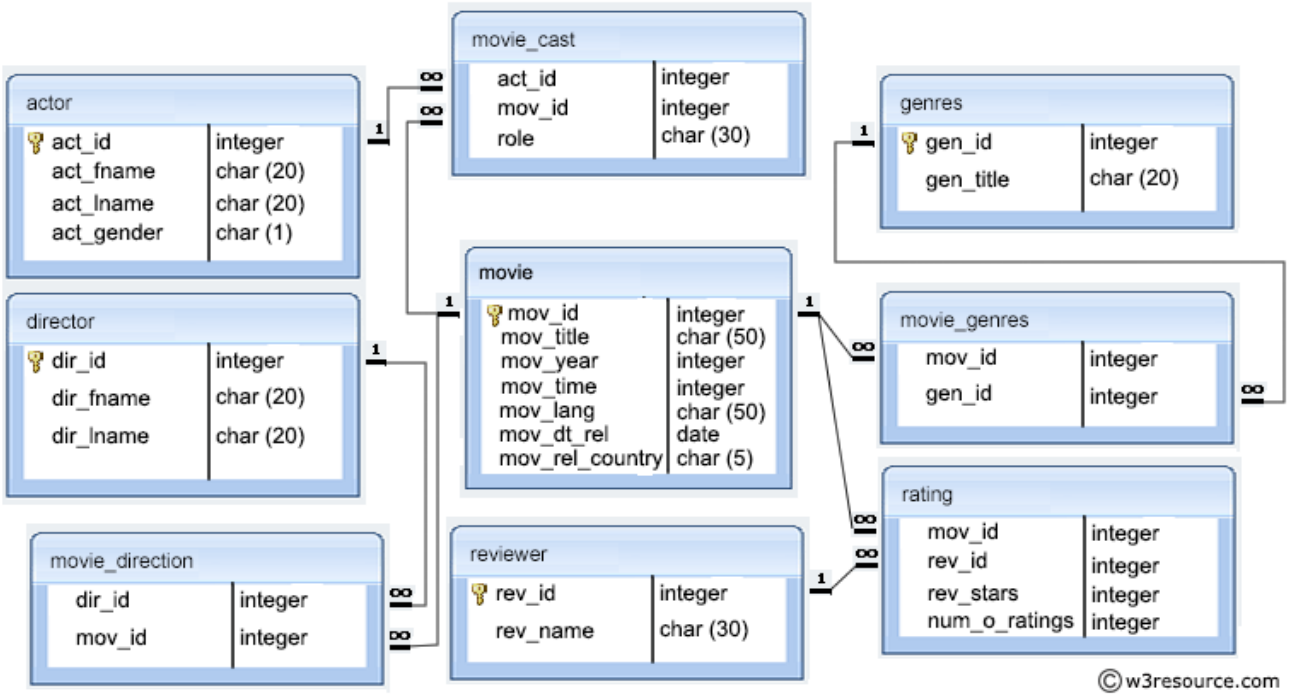
Day -13

W3Resource SQL Movie Database



Database can be downloaded from w3resources and also from my Github account

https://github.com/im-amit-kumar/100-DAYS-OF-DATA-SCIENCE/tree/main/Day-13

13.From the following tables, write a SQL query to get the reviewer name, movie title, and stars in an order that reviewer name will come first, then by movie title, and lastly by number of stars.

```
select rev_name , mov_title , rev_stars
from reviewer
natural join rating
natural join movie
where rev_name is not null
```

```
order by  rev_name , mov_title , rev_stars;
```

14. From the following tables, write a SQL query to find those movies that have at least one rating and received highest number of stars. Sort the result-set on movie title. Return movie title and maximum review stars.

```
select mov_title , max(rev_stars)
from movie m
join rating r
on m.mov_id = r.mov_id
group by mov_title
having max(rev_stars) > 0
order by mov_title ;
```

15. From the following tables, write a SQL query to find those movies, which have received ratings. Return movie title, director first name, director last name and review stars.

```
select mov_title , dir_fname , dir_lname , rev_Stars
from movie
join movie_direction using (mov_id)
join director using(dir_id)
join rating using(mov_id)
where rev_stars is not null ;
```

16. Write a query in SQL to find the movie title, actor first and last name, and the role for those movies where one or more actors acted in two or more movies.

```
SELECT mov_title, act_fname, act_lname, role
FROM movie
JOIN movie_cast
  ON movie_cast.mov_id=movie.mov_id
JOIN actor
  ON movie_cast.act_id=actor.act_id
WHERE actor.act_id IN (
SELECT act_id
FROM movie_cast
GROUP BY act_id HAVING COUNT(*)>=2);
```

17. From the following table, write a SQL query to find the actor whose first name is ‘Claire’ and last name is 'Danes'. Return director first name, last name, movie title, actor first name and last name, role.

```
SELECT dir_fname, dir_lname, mov_title, act_fname, act_lname, role
FROM actor ac
JOIN movie_cast mc
  ON ac.act_id= mc.act_id
JOIN movie_direction md
  ON mc.mov_id=md.mov_id
JOIN director d
  ON md.dir_id=d.dir_id
```

```
JOIN movie m1
  ON m1.mov_id=md.mov_id
WHERE act_fname='Claire'
  AND act_lname='Danes';
```

18. From the following tables, write a SQL query to find those actors who have directed their movies. Return actor first name, last name, movie title and role.

```
SELECT act_fname, act_lname, mov_title, role
FROM actor ac
JOIN movie_cast mc
  ON ac.act_id=mc.act_id
JOIN movie_direction md
  ON mc.mov_id=md.mov_id
JOIN director d
  ON md.dir_id=d.dir_id
JOIN movie m
  ON m.mov_id=md.mov_id
WHERE act_fname=dir_fname
  AND act_lname=dir_lname;
```

19. From the following tables, write a SQL query to find the cast list of the movie 'Chinatown'. Return first name, last name.

```
select a.act_fname , a.act_lname
```

```
from

movie_cast c

join actor a on

c.act_id = a.act_id

where mov_id = (
select mov_id from movie
 where mov_title ='Chinatown');
```

20. From the following tables, write a SQL query to find those movies where actor's first name is 'Harrison' and last name is 'Ford'. Return movie title.

```
select m.mov_title
from movie m
join movie_cast c
on m.mov_id = c.mov_id
where c.act_id in (
select act_id from actor where act_fname = 'Harrison'
and act_lname = 'Ford'
```

);

21. From the following tables, write a SQL query to find the highest-rated movies. Return movie title, movie year, review stars and releasing country.

```
select mov_title , mov_year , rev_stars , mov_rel_country

from movie

natural join rating

where rev_stars = (

select max(rev_stars) from rating);
```

22. From the following tables, write a SQL query to find the highest-rated 'Mystery Movies'. Return the title, year, and rating.

```
select mov_title , mov_year , rev_stars

from movie

natural join movie_genres

natural join genres

natural join rating

where gen_title = 'Mystery' AND rev_stars >= ALL (


select rev_stars

from rating
```

```
natural join movie_genres

natural join genres

where gen_title =' Mystery');
```

23. From the following tables, write a SQL query to find the years when most of the 'Mystery Movies' produced. Count the number of generic title and compute their average rating. Group the result set on movie release year, generic title. Return movie year, generic title, number of generic title and average rating.

```
select mov_year , gen_title , count(gen_title) , avg(rev_stars)

from movie

natural join movie_genres

natural join genres

natural join rating

where gen_title = 'Mystery'

group by mov_year  , gen_title ;
```

24. From the following tables, write a query in SQL to generate a report, which contain the fields movie title, name of the female actor, year of the movie, role, movie genres, the director, date of release, and rating of that movie.

```
select mov_title , act_fname , act_lname

mov_year , role , gen_title , dir_fname , dir_lname, mov_Dt_rel, rev_stars
```

```
from movie
natural join movie_cast
natural join actor
natural join movie_genres
natural join genres
natural join movie_direction
natural join director
natural join rating
where act_gender ='F';
```