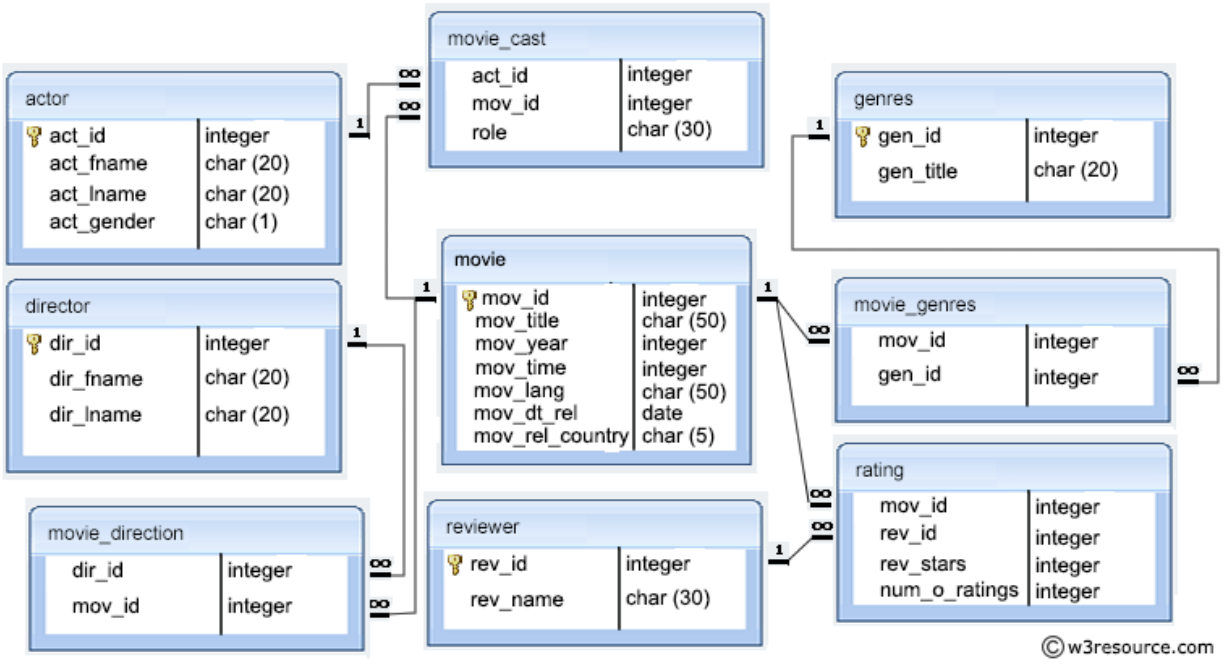
DAY-15

SQL W3 RESOURCE



Movie Database

Database can be downloaded from w3 resource and also from my Github

https://github.com/im-amit-kumar/100-DAYS-OF-DATA-SCIENCE/tree/main/DAY-15

9. From the following tables, write a SQL query to find those movies, which reviewed by a reviewer and got a rating. Sort the result-set in ascending order by reviewer name, movie title, review Stars. Return reviewer name, movie title, review Stars.

```
select rev_name , mov_title , rev_stars

from reviewer a , movie b , rating c

where a.rev_id = c.rev_id
```

```
and b.mov_id = c.mov_id
and a.rev_name is not null
and c.rev_stars is not null
order by rev_name , mov_title , rev_stars;
```

10. From the following tables, write a SQL query to find those reviewers who rated more than one movie. Group the result set on reviewer's name, movie title. Return reviewer's name, movie title.

```
select rev_name , mov_title
from reviewer a , movie b , rating c , rating d
where a.rev_id = c.rev_id
and c.mov_id = b.mov_id
and c.rev_id = d.rev_id
group by rev_name , mov_title
having count(*) > 1;
```

11. From the following tables, write a SQL query to find those movies, which have received highest number of stars. Group the result set on movie title and sorts the result-set in ascending order by movie title. Return movie title and maximum number of review stars.

```
select mov_title , max(rev_stars)
from rating a, movie b
where a.mov_id = b.mov_id
and a.rev_stars is not null
```

```
group by mov_title
order by mov_title;
```

12. **From the following table, write a SQL query to find all reviewers who rated the movie ‘American Beauty’. Return reviewer name.**

```
select rev_name
from reviewer a, rating b , movie c
where a.rev_id = b.rev_id
and b.mov_id = c.mov_id
and mov_title= 'American Beauty';;
```

13. From the following tables, write a SQL query to find the movies, which have reviewed by any reviewer body except by 'Paul Monks'. Return movie title.

```
select mov_title from movie
where mov_id in (

select mov_id from rating where rev_id not in
(
select rev_id from reviewer where rev_name ='Paul Marks'
)
);
```

14. From the following tables, write a SQL query to find the lowest rated movies. Return reviewer name, movie title, and number of stars for those movies.

```
select rev_name , mov_title , rev_stars
from reviewer a, movie b, rating c
where rev_stars = (
select min(rev_stars ) from rating
)
and c.rev_id = a.rev_id
and c.mov_id = b.mov_id;
```

15. **From the following tables, write a SQL query to find the movies directed by 'James Cameron'. Return movie title.**

```
SELECT mov_title
FROM movie
WHERE mov_id IN (
SELECT mov_id
FROM movie_direction
WHERE dir_id IN (
SELECT dir_id
FROM director
WHERE dir_fname = 'James' AND dir_lname='Cameron'
));
```

16. **Write a query in SQL to find the name of those movies where one or more actors acted in two or more movies.**

```
SELECT mov_title   FROM movie

WHERE mov_id  IN  ( SELECT mov_id

FROM movie_cast  WHERE act_id IN (

SELECT act_id  FROM actor

WHERE act_id IN (  SELECT act_id

FROM movie_cast GROUP BY act_id

HAVING COUNT(act_id)>1)));
```