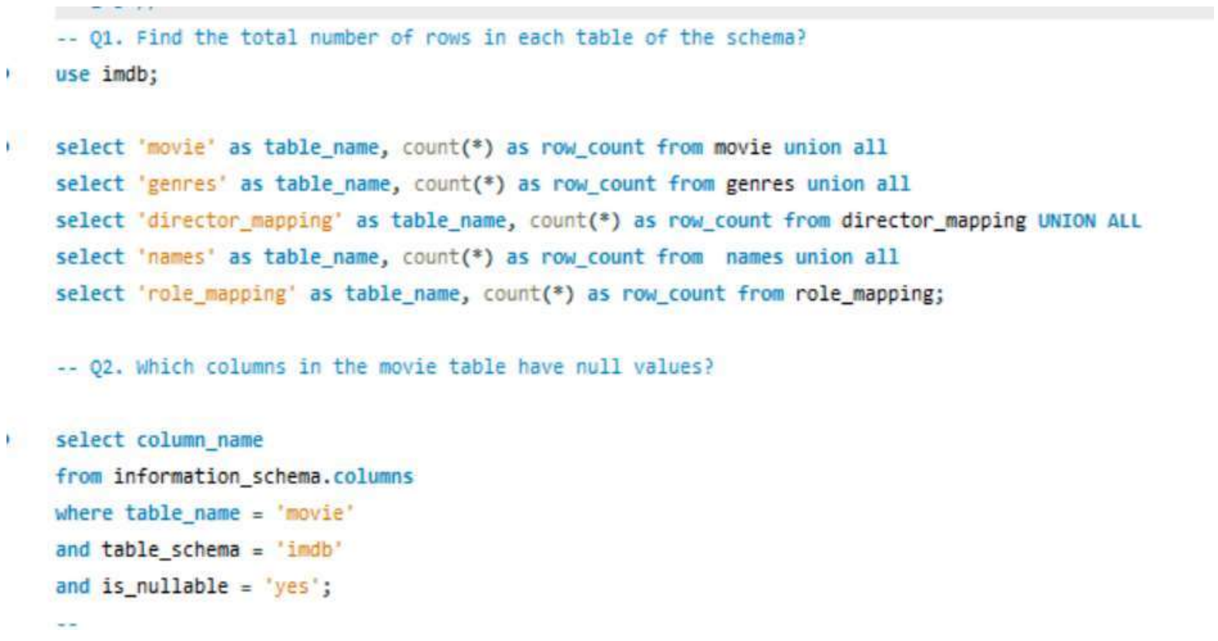
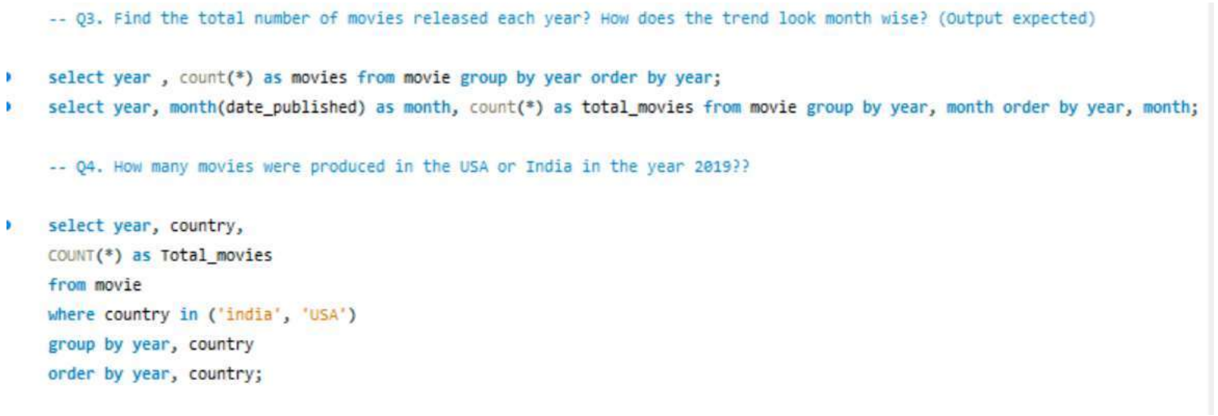
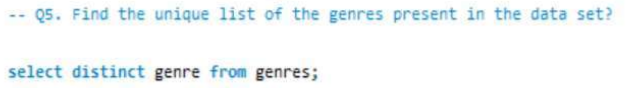
MySQL PROJECT: MOVIE ANALYSIS FOR RSVP BASED ON IMDB SCORES



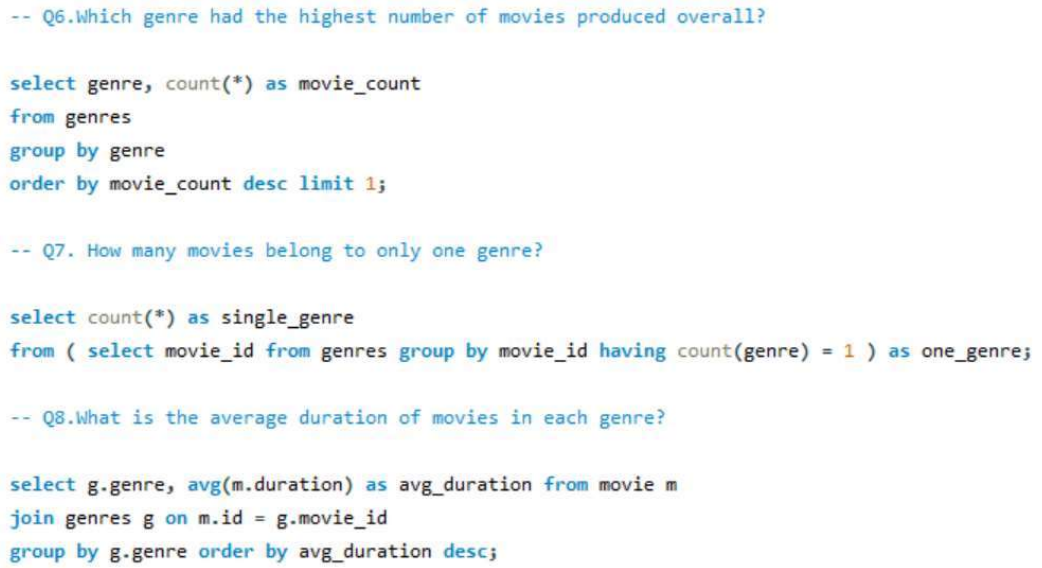
```
-- Q1. Find the total number of rows in each table of the schema?
use imdb;

select 'movie' as table_name, count(*) as row_count from movie union all
select 'genres' as table_name, count(*) as row_count from genres union all
select 'director_mapping' as table_name, count(*) as row_count from director_mapping UNION ALL
select 'names' as table_name, count(*) as row_count from  names union all
select 'role_mapping' as table_name, count(*) as row_count from role_mapping;

-- Q2. Which columns in the movie table have null values?

select column_name
from information_schema.columns
where table_name = 'movie'
and table_schema = 'imdb'
and is_nullable = 'yes';
--
```



```
-- Q3. Find the total number of movies released each year? How does the trend look month wise? (Output expected)

select year , count(*) as movies from movie group by year order by year;
select year, month(date_published) as month, count(*) as total_movies from movie group by year, month order by year, month;

-- Q4. How many movies were produced in the USA or India in the year 2019??

select year, country,
COUNT(*) as Total_movies
from movie
where country in ('india', 'USA')
group by year, country
order by year, country;
```

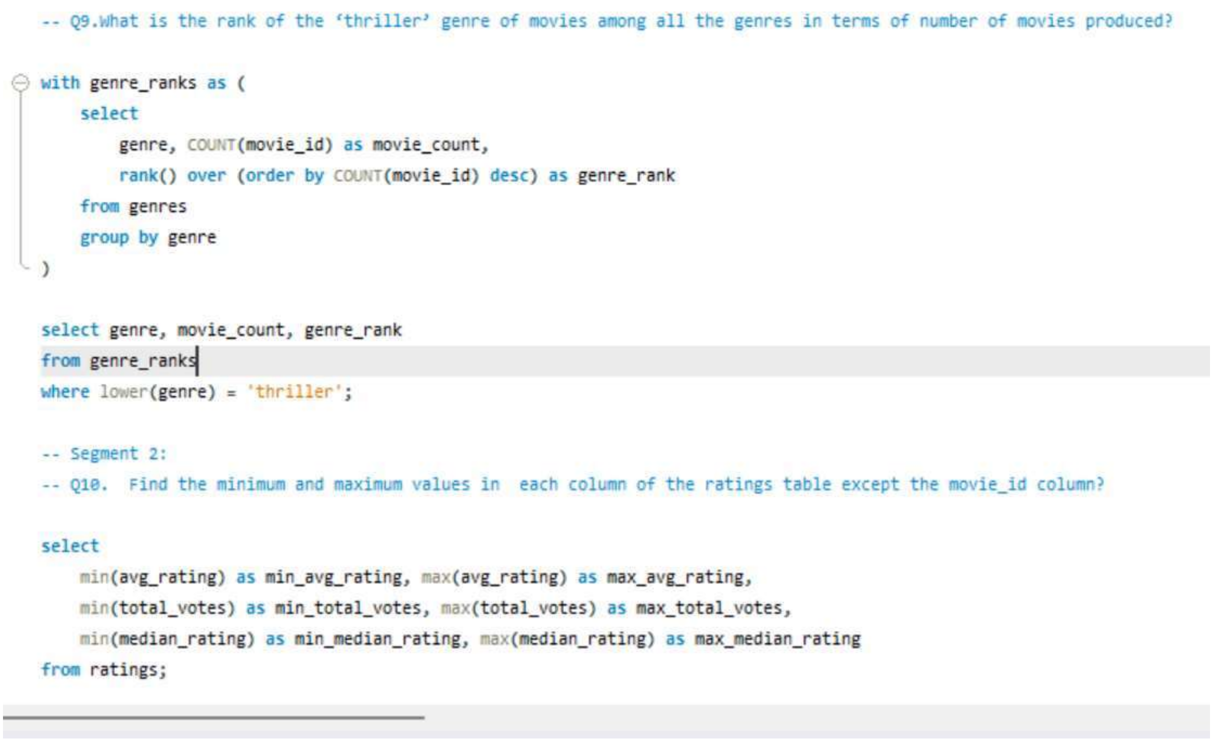


```
-- Q5. Find the unique list of the genres present in the data set?

select distinct genre from genres;
```

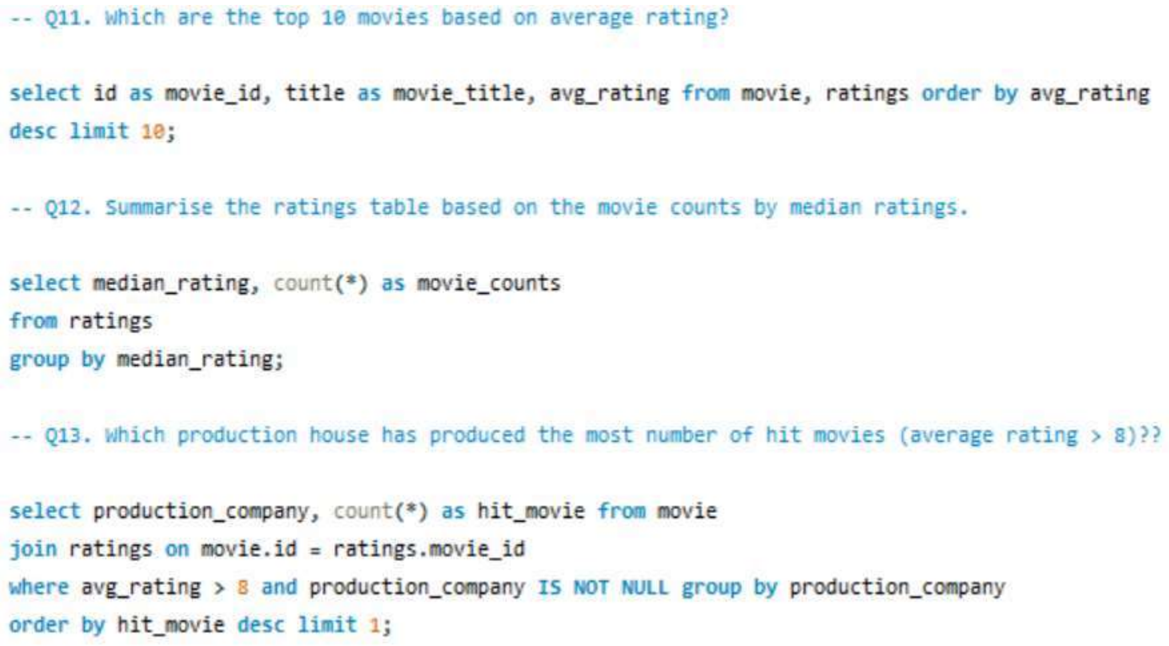


```
-- Q6.Which genre had the highest number of movies produced overall?

select genre, count(*) as movie_count
from genres
group by genre
order by movie_count desc limit 1;

-- Q7. How many movies belong to only one genre?

select count(*) as single_genre
from ( select movie_id from genres group by movie_id having count(genre) = 1 ) as one_genre;

-- Q8.What is the average duration of movies in each genre?

select g.genre, avg(m.duration) as avg_duration from movie m
join genres g on m.id = g.movie_id
group by g.genre order by avg_duration desc;
```

MySQL PROJECT: MOVIE ANALYSIS FOR RSVP BASED ON IMDB SCORES

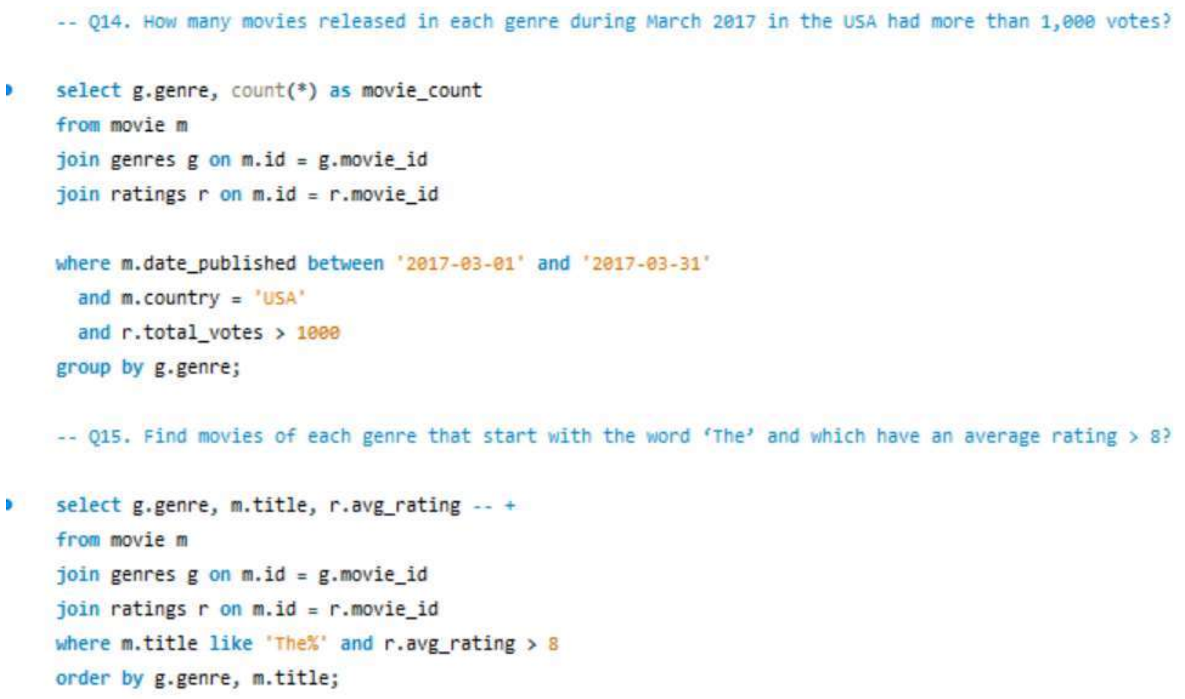


```
-- Q9.What is the rank of the ‘thriller’ genre of movies among all the genres in terms of number of movies produced?

with genre_ranks as (
    select
        genre, COUNT(movie_id) as movie_count,
        rank() over (order by COUNT(movie_id) desc) as genre_rank
    from genres
    group by genre
)

select genre, movie_count, genre_rank
from genre_ranks
where lower(genre) = 'thriller';

-- Segment 2:
-- Q10.  Find the minimum and maximum values in  each column of the ratings table except the movie_id column?

select
    min(avg_rating) as min_avg_rating, max(avg_rating) as max_avg_rating,
    min(total_votes) as min_total_votes, max(total_votes) as max_total_votes,
    min(median_rating) as min_median_rating, max(median_rating) as max_median_rating
from ratings;
```

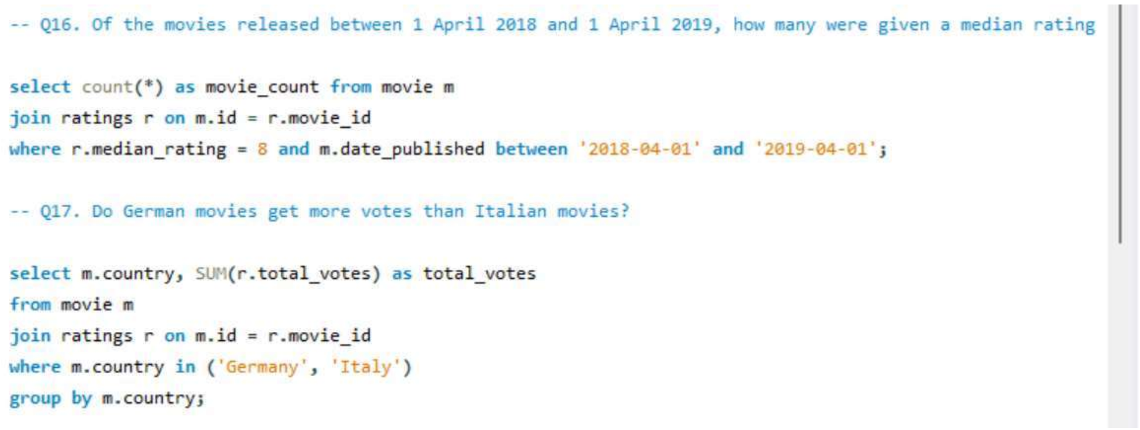


```
-- Q11. Which are the top 10 movies based on average rating?

select id as movie_id, title as movie_title, avg_rating from movie, ratings order by avg_rating
desc limit 10;

-- Q12. Summarise the ratings table based on the movie counts by median ratings.

select median_rating, count(*) as movie_counts
from ratings
group by median_rating;

-- Q13. Which production house has produced the most number of hit movies (average rating > 8)??

select production_company, count(*) as hit_movie from movie
join ratings on movie.id = ratings.movie_id
where avg_rating > 8 and production_company IS NOT NULL group by production_company
order by hit_movie desc limit 1;
```

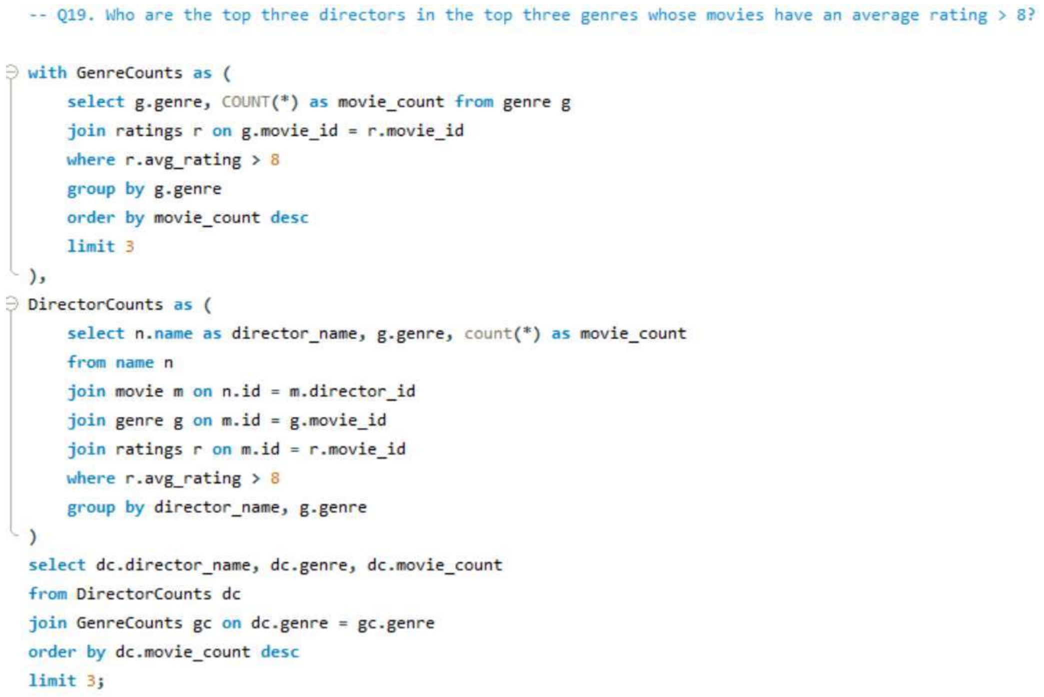
MySQL PROJECT: MOVIE ANALYSIS FOR RSVP BASED ON IMDB SCORES



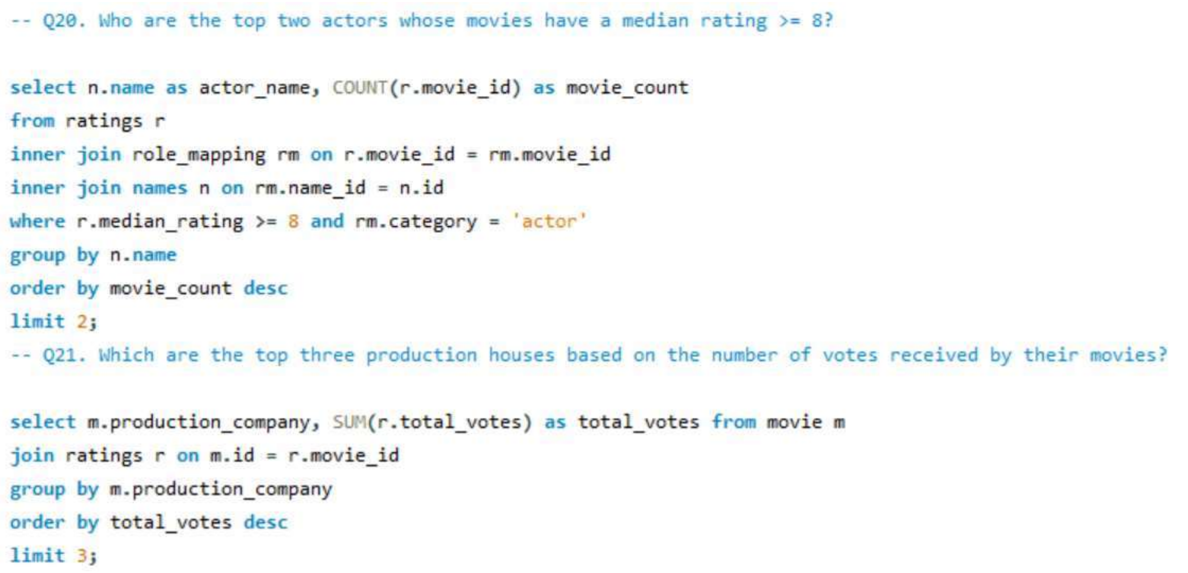
```
-- Q14. How many movies released in each genre during March 2017 in the USA had more than 1,000 votes?

select g.genre, count(*) as movie_count
from movie m
join genres g on m.id = g.movie_id
join ratings r on m.id = r.movie_id

where m.date_published between '2017-03-01' and '2017-03-31'
  and m.country = 'USA'
  and r.total_votes > 1000
group by g.genre;

-- Q15. Find movies of each genre that start with the word 'The' and which have an average rating > 8?

select g.genre, m.title, r.avg_rating -- +
from movie m
join genres g on m.id = g.movie_id
join ratings r on m.id = r.movie_id
where m.title like 'The%' and r.avg_rating > 8
order by g.genre, m.title;
```



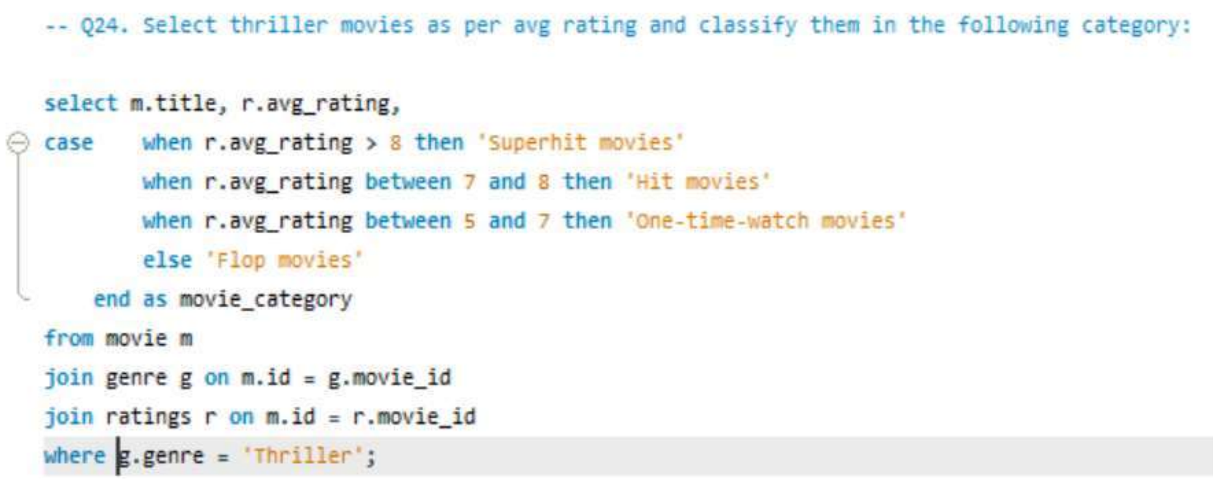
```
-- Q16. Of the movies released between 1 April 2018 and 1 April 2019, how many were given a median rating

select count(*) as movie_count from movie m
join ratings r on m.id = r.movie_id
where r.median_rating = 8 and m.date_published between '2018-04-01' and '2019-04-01';

-- Q17. Do German movies get more votes than Italian movies?

select m.country, SUM(r.total_votes) as total_votes
from movie m
join ratings r on m.id = r.movie_id
where m.country in ('Germany', 'Italy')
group by m.country;
```



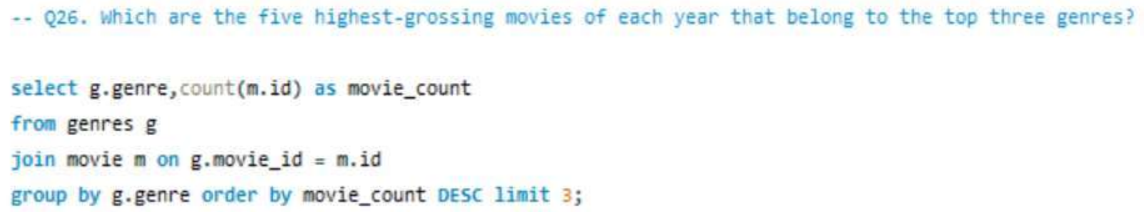
```
-- Q19. Who are the top three directors in the top three genres whose movies have an average rating > 8?

with GenreCounts as (
    select g.genre, COUNT(*) as movie_count from genre g
    join ratings r on g.movie_id = r.movie_id
    where r.avg_rating > 8
    group by g.genre
    order by movie_count desc
    limit 3
),
DirectorCounts as (
    select n.name as director_name, g.genre, count(*) as movie_count
    from name n
    join movie m on n.id = m.director_id
    join genre g on m.id = g.movie_id
    join ratings r on m.id = r.movie_id
    where r.avg_rating > 8
    group by director_name, g.genre
)
select dc.director_name, dc.genre, dc.movie_count
from DirectorCounts dc
join GenreCounts gc on dc.genre = gc.genre
order by dc.movie_count desc
limit 3;
```



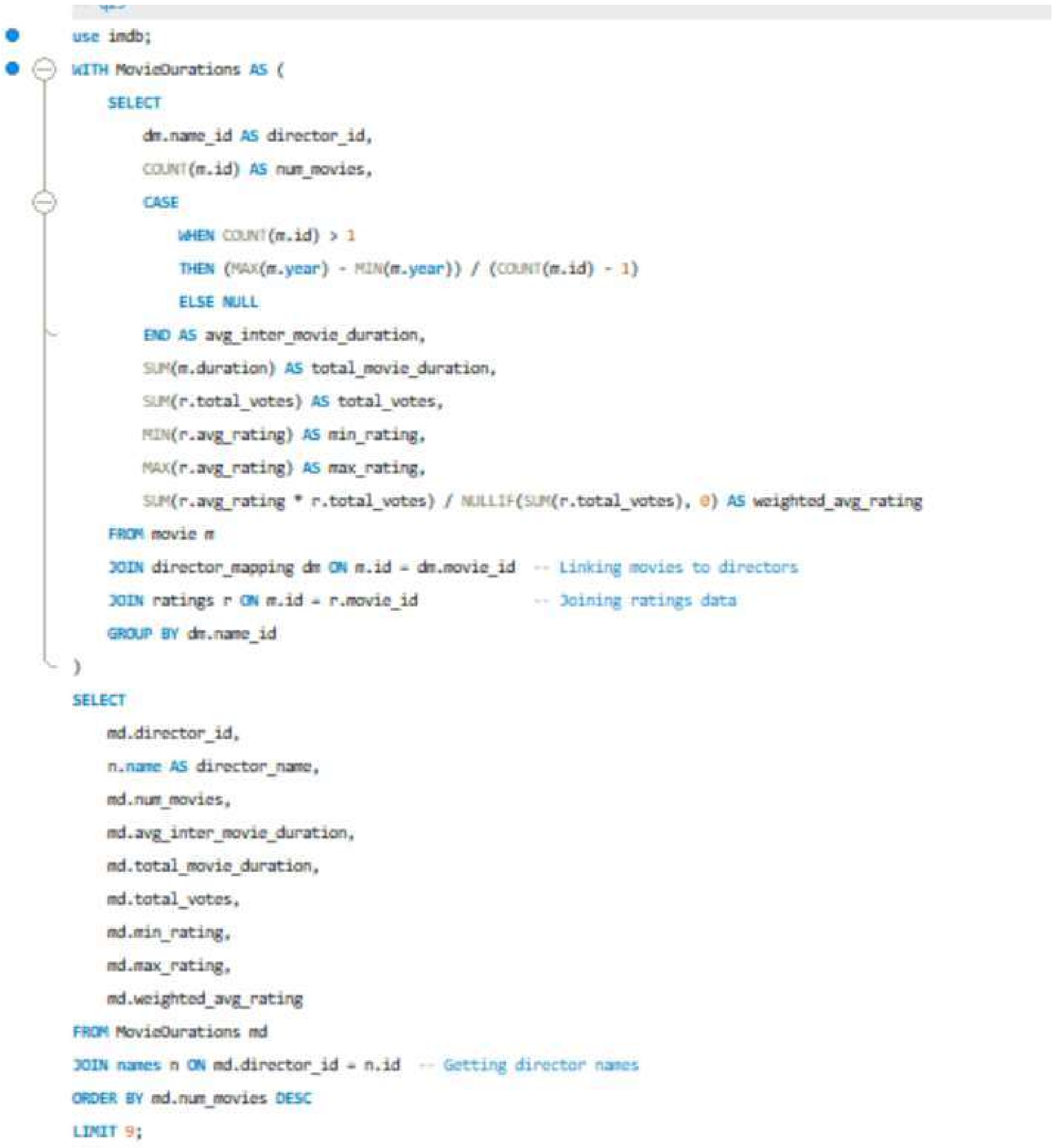
```
-- Q20. Who are the top two actors whose movies have a median rating >= 8?

select n.name as actor_name, COUNT(r.movie_id) as movie_count
from ratings r
inner join role_mapping rm on r.movie_id = rm.movie_id
inner join names n on rm.name_id = n.id
where r.median_rating >= 8 and rm.category = 'actor'
group by n.name
order by movie_count desc
limit 2;
-- Q21. Which are the top three production houses based on the number of votes received by their movies?

select m.production_company, SUM(r.total_votes) as total_votes from movie m
join ratings r on m.id = r.movie_id
group by m.production_company
order by total_votes desc
limit 3;
```



```
-- Q24. Select thriller movies as per avg rating and classify them in the following category:

select m.title, r.avg_rating,
case    when r.avg_rating > 8 then 'Superhit movies'
        when r.avg_rating between 7 and 8 then 'Hit movies'
        when r.avg_rating between 5 and 7 then 'One-time-watch movies'
        else 'Flop movies'
    end as movie_category
from movie m
join genre g on m.id = g.movie_id
join ratings r on m.id = r.movie_id
where g.genre = 'Thriller';
```



```
-- Q26. Which are the five highest-grossing movies of each year that belong to the top three genres?

select g.genre,count(m.id) as movie_count
from genres g
join movie m on g.movie_id = m.id
group by g.genre order by movie_count DESC limit 3;
```



```
use imdb;
WITH MovieDurations AS (
    SELECT
        dm.name_id AS director_id,
        COUNT(m.id) AS num_movies,
        CASE
            WHEN COUNT(m.id) > 1
            THEN (MAX(m.year) - MIN(m.year)) / (COUNT(m.id) - 1)
            ELSE NULL
        END AS avg_inter_movie_duration,
        SUM(m.duration) AS total_movie_duration,
        SUM(r.total_votes) AS total_votes,
        MIN(r.avg_rating) AS min_rating,
        MAX(r.avg_rating) AS max_rating,
        SUM(r.avg_rating * r.total_votes) / NULLIF(SUM(r.total_votes), 0) AS weighted_avg_rating
    FROM movie m
    JOIN director_mapping dm ON m.id = dm.movie_id    -- Linking movies to directors
    JOIN ratings r ON m.id = r.movie_id               -- Joining ratings data
    GROUP BY dm.name_id
)
SELECT
    md.director_id,
    n.name AS director_name,
    md.num_movies,
    md.avg_inter_movie_duration,
    md.total_movie_duration,
    md.total_votes,
    md.min_rating,
    md.max_rating,
    md.weighted_avg_rating
FROM MovieDurations md
JOIN names n ON md.director_id = n.id   -- Getting director names
ORDER BY md.num_movies DESC
LIMIT 9;
```