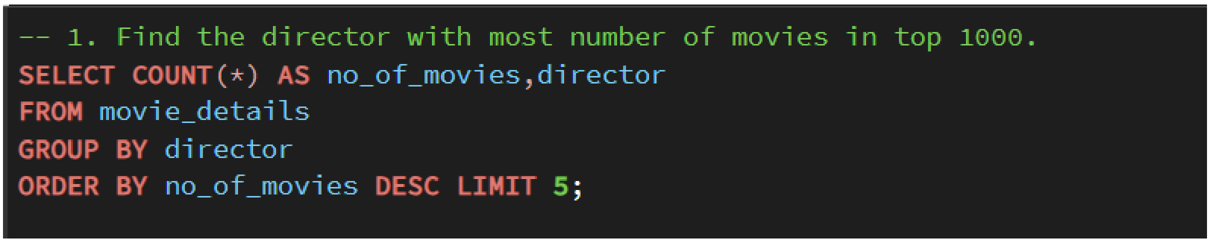
IMDB Movie Analysis Report

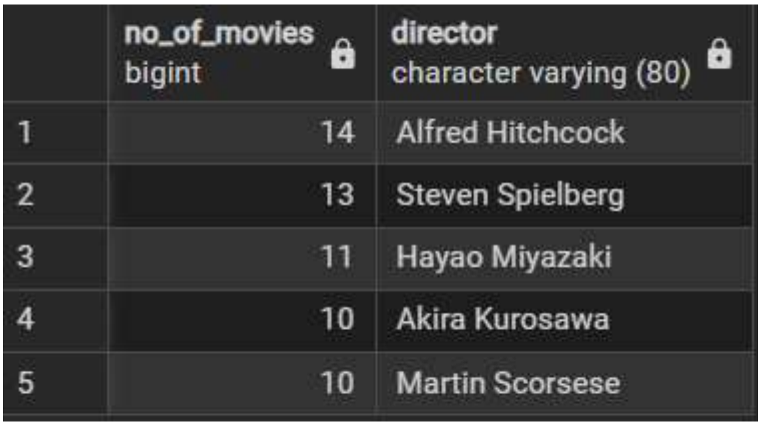
Data Analysis and Insights:

1. Find the director with most no. of movies in top 1000.



```sql
-- 1. Find the director with most number of movies in top 1000.
SELECT COUNT(*) AS no_of_movies,director
FROM movie_details
GROUP BY director
ORDER BY no_of_movies DESC LIMIT 5;
```

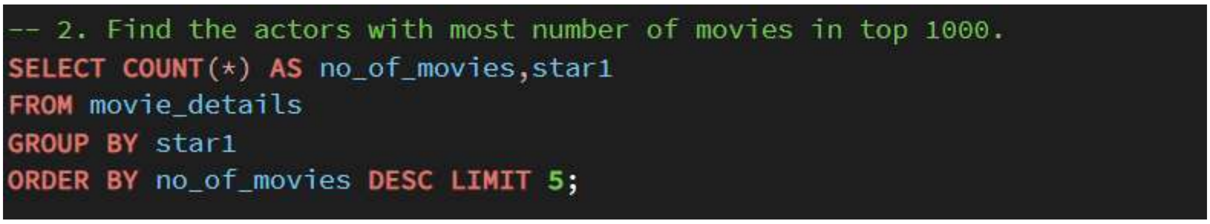
Output:



	no_of_movies bigint	director character varying (80)
1	14	Alfred Hitchcock
2	13	Steven Spielberg
3	11	Hayao Miyazaki
4	10	Akira Kurosawa
5	10	Martin Scorsese

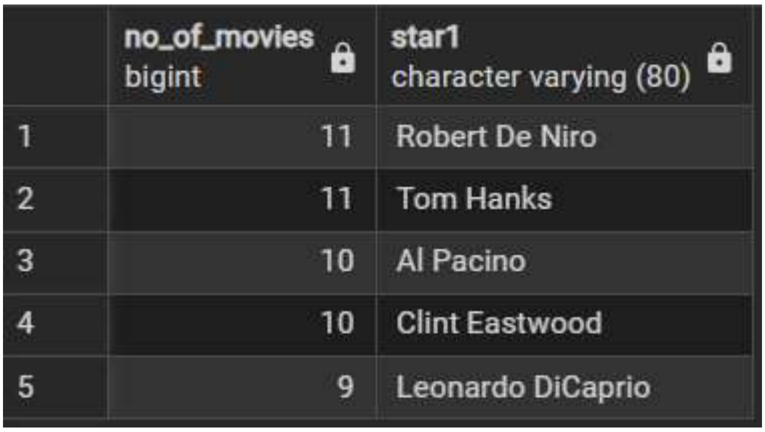
With 14 movies, **Alfred Hitchcock** has the most entries in the top 1000, followed by **Steven Spielberg** and **Hayao Miyazaki.**

2. Find the actor with most no. of movies in top 1000.



```
-- 2. Find the actors with most number of movies in top 1000.
SELECT COUNT(*) AS no_of_movies,star1
FROM movie_details
GROUP BY star1
ORDER BY no_of_movies DESC LIMIT 5;
```

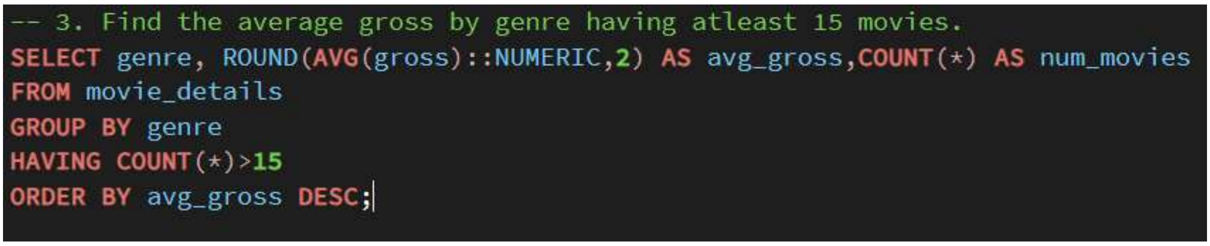
Output:



	no_of_movies bigint	star1 character varying (80)
1	11	Robert De Niro
2	11	Tom Hanks
3	10	Al Pacino
4	10	Clint Eastwood
5	9	Leonardo DiCaprio

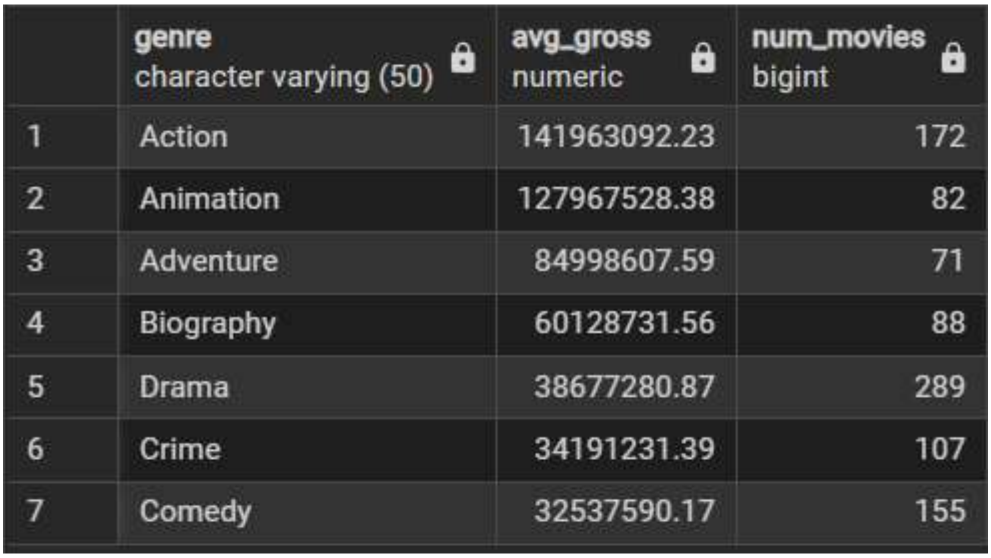
Robert De Niro and **Tom Hanks** each have 11 movies in the top 1000, the most of any actor.

3. Find the average gross by genre having at least 15 movies.



```
-- 3. Find the average gross by genre having atleast 15 movies.
SELECT genre, ROUND(AVG(gross)::NUMERIC,2) AS avg_gross,COUNT(*) AS num_movies
FROM movie_details
GROUP BY genre
HAVING COUNT(*)>15
ORDER BY avg_gross DESC;
```

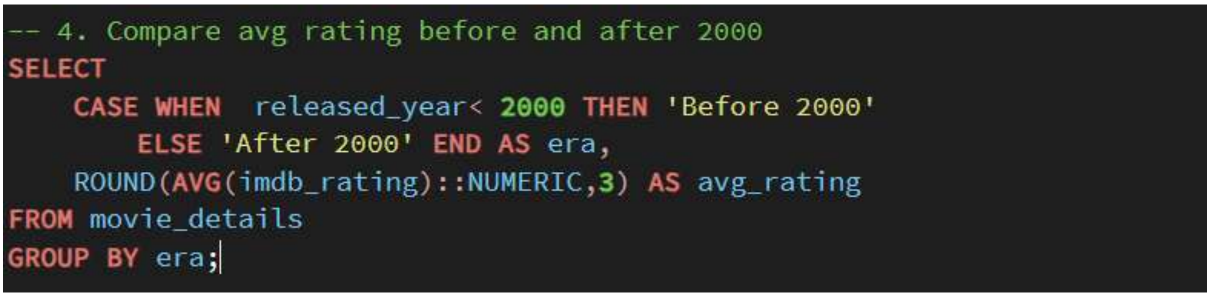
Output:



	genre character varying (50)	avg_gross numeric	num_movies bigint
1	Action	141963092.23	172
2	Animation	127967528.38	82
3	Adventure	84998607.59	71
4	Biography	60128731.56	88
5	Drama	38677280.87	289
6	Crime	34191231.39	107
7	Comedy	32537590.17	155

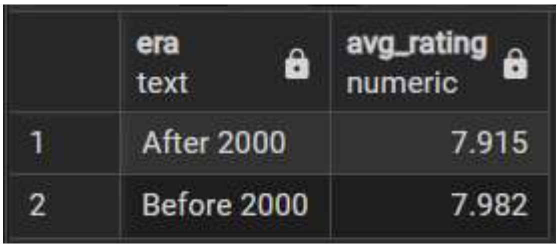
Action movies, with at least 15 entries, have the highest average gross, followed by **Animation** and **Adventure**.

4. Compute average rating before and after 2000.



```
-- 4. Compare avg rating before and after 2000
SELECT
    CASE WHEN  released_year< 2000 THEN 'Before 2000'
        ELSE 'After 2000' END AS era,
    ROUND(AVG(imdb_rating)::NUMERIC,3) AS avg_rating
FROM movie_details
GROUP BY era;
```

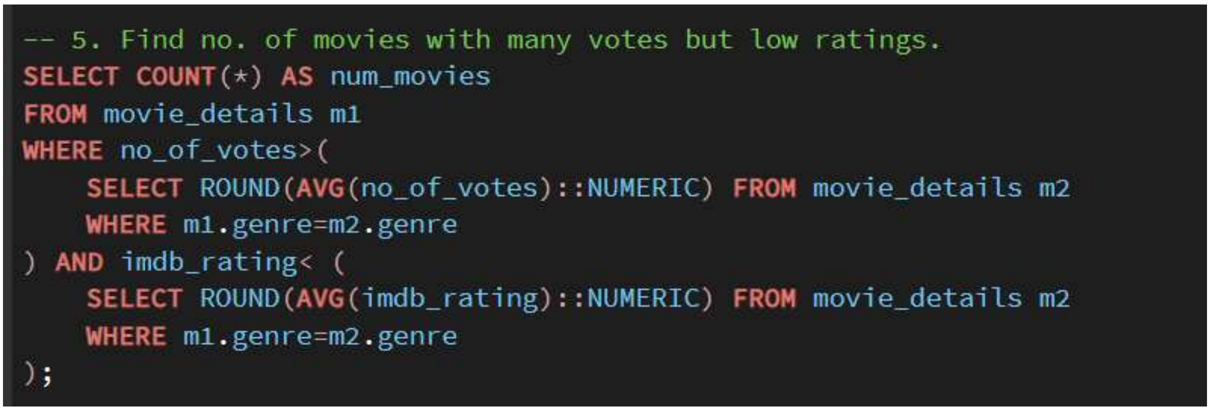
Output:



	era text	avg_rating numeric
1	After 2000	7.915
2	Before 2000	7.982

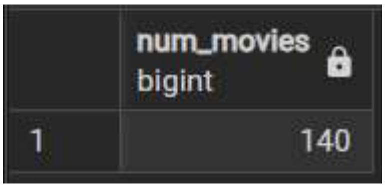
Movies **released after 2000** have a higher average rating than those **released before 2000**.

5. Find no. of movies with many votes but low ratings.



```
-- 5. Find no. of movies with many votes but low ratings.
SELECT COUNT(*) AS num_movies
FROM movie_details m1
WHERE no_of_votes>(
    SELECT ROUND(AVG(no_of_votes)::NUMERIC) FROM movie_details m2
    WHERE m1.genre=m2.genre
) AND imdb_rating< (
    SELECT ROUND(AVG(imdb_rating)::NUMERIC) FROM movie_details m2
    WHERE m1.genre=m2.genre
);
```

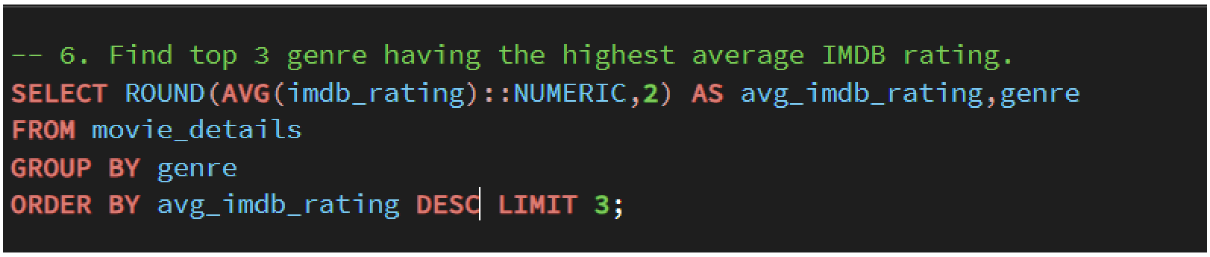
Output:



	num_movies bigint
1	140

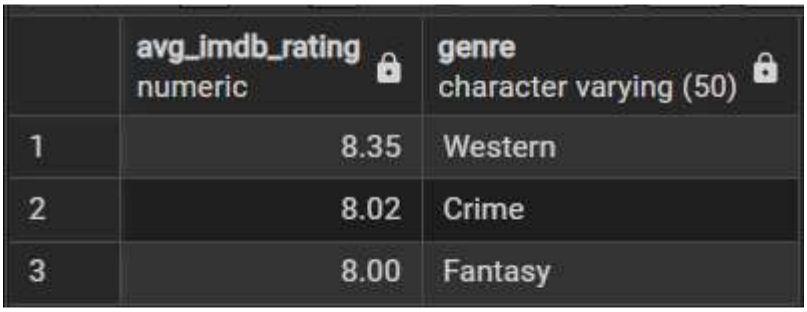
140 of the 999 movies are popular among viewers, having high vote counts. However, their ratings are lower than the average for their respective genres.

6. Find top 3 genre has the highest rated IMDB ratings.



```sql
-- 6. Find top 3 genre having the highest average IMDB rating.
SELECT ROUND(AVG(imdb_rating)::NUMERIC,2) AS avg_imdb_rating,genre
FROM movie_details
GROUP BY genre
ORDER BY avg_imdb_rating DESC LIMIT 3;
```

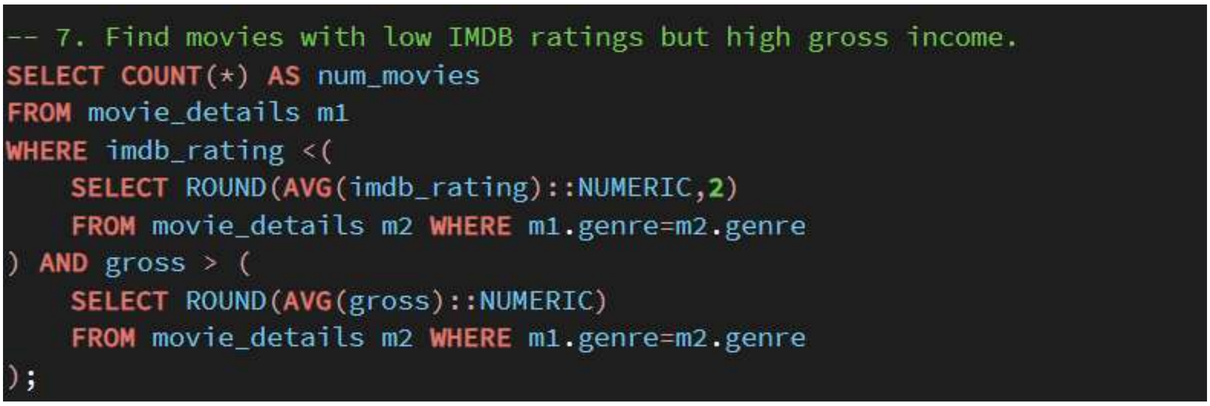
Output:



	avg_imdb_rating numeric	genre character varying (50)
1	8.35	Western
2	8.02	Crime
3	8.00	Fantasy

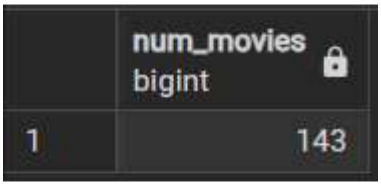
The **Western** genre has the highest average IMDB rating at 8.35, followed by **Crime** and **Fantasy.**

7. Find movies with low IMDB rating but high gross income.



```
-- 7. Find movies with low IMDB ratings but high gross income.
SELECT COUNT(*) AS num_movies
FROM movie_details m1
WHERE imdb_rating <(
    SELECT ROUND(AVG(imdb_rating)::NUMERIC,2)
    FROM movie_details m2 WHERE m1.genre=m2.genre
) AND gross > (
    SELECT ROUND(AVG(gross)::NUMERIC)
    FROM movie_details m2 WHERE m1.genre=m2.genre
);
```

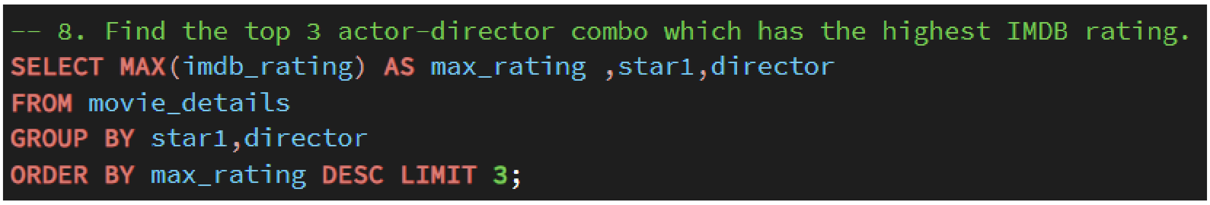
Output:



	num_movies bigint
1	143

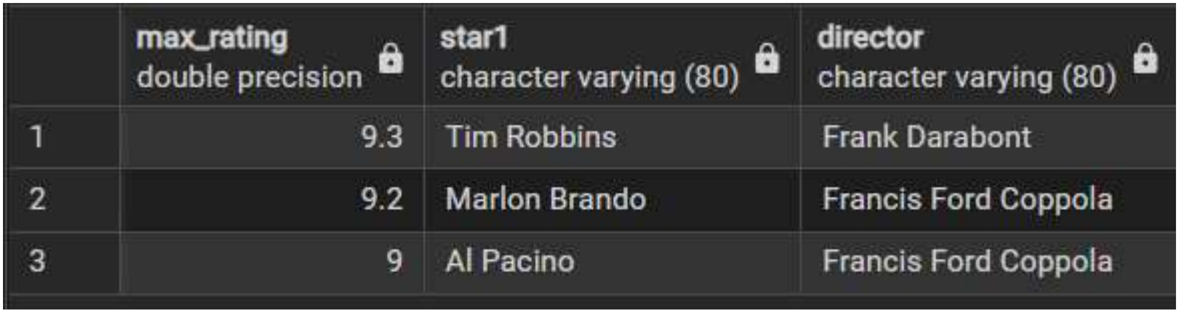
Despite having lower IMDB ratings than their genre's average, **143 of the 999 movies** earned higher gross income than their genre's average.

8. Find the top 3 actor-director combo which has the highest IMDB rating.



```
-- 8. Find the top 3 actor-director combo which has the highest IMDB rating.
SELECT MAX(imdb_rating) AS max_rating ,star1,director
FROM movie_details
GROUP BY star1,director
ORDER BY max_rating DESC LIMIT 3;
```

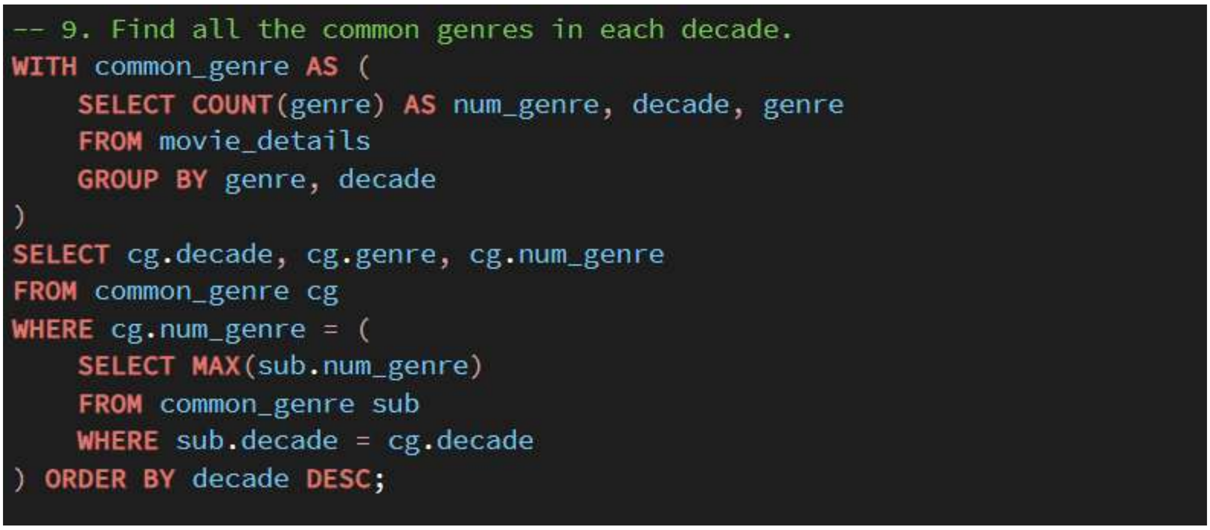
Output:



	max_rating double precision	star1 character varying (80)	director character varying (80)
1	9.3	Tim Robbins	Frank Darabont
2	9.2	Marlon Brando	Francis Ford Coppola
3	9	Al Pacino	Francis Ford Coppola

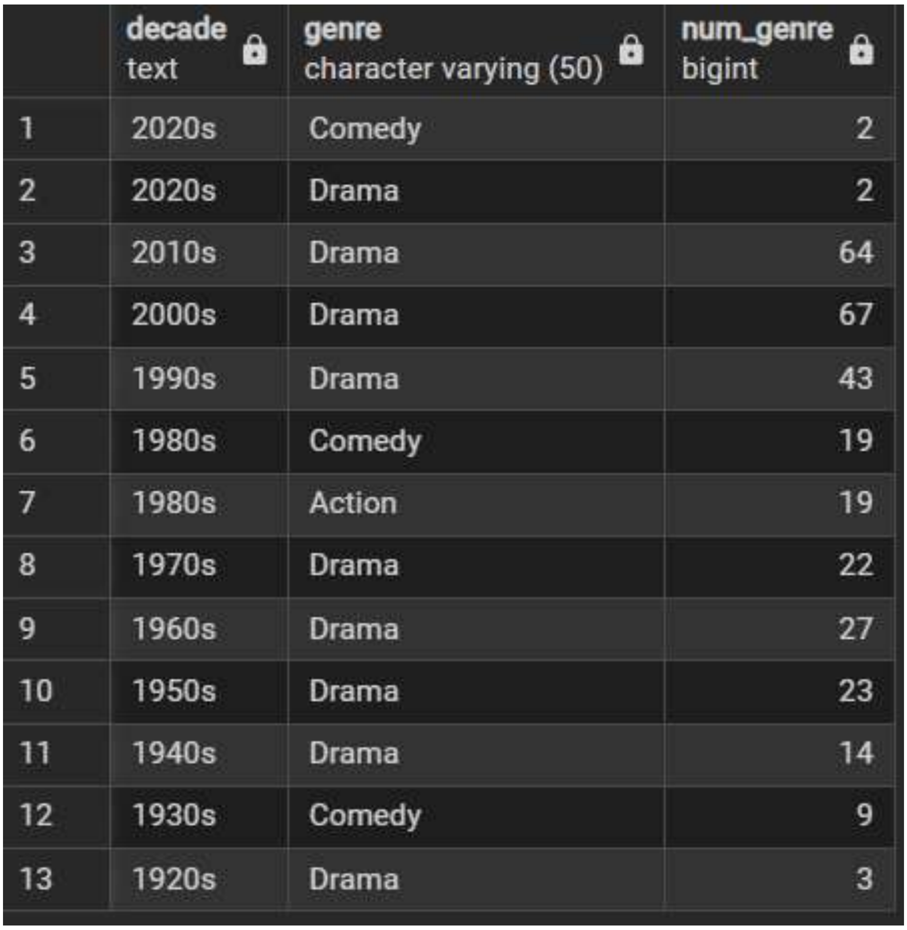
The **Tim Robbins and Frank Darabont** "combo" achieves the highest IMDB rating of 9.3, followed by the **Marlon Brando and Francis Ford Coppola** pairing, and the **Al Pacino** and **Francis Ford Coppola** pairing.

9. **Find all the common genres in each decade.**



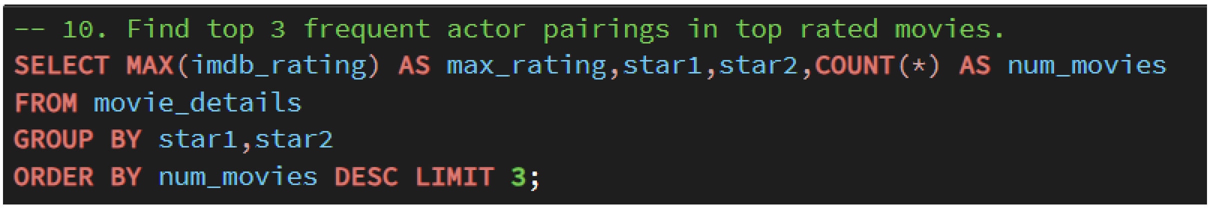
```
-- 9. Find all the common genres in each decade.
WITH common_genre AS (
    SELECT COUNT(genre) AS num_genre, decade, genre
    FROM movie_details
    GROUP BY genre, decade
)
SELECT cg.decade, cg.genre, cg.num_genre
FROM common_genre cg
WHERE cg.num_genre = (
    SELECT MAX(sub.num_genre)
    FROM common_genre sub
    WHERE sub.decade = cg.decade
) ORDER BY decade DESC;
```

Output:



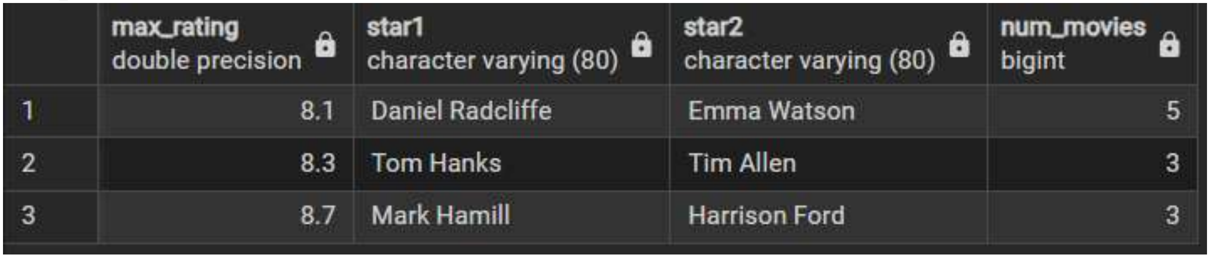
	decade text	genre character varying (50)	num_genre bigint
1	2020s	Comedy	2
2	2020s	Drama	2
3	2010s	Drama	64
4	2000s	Drama	67
5	1990s	Drama	43
6	1980s	Comedy	19
7	1980s	Action	19
8	1970s	Drama	22
9	1960s	Drama	27
10	1950s	Drama	23
11	1940s	Drama	14
12	1930s	Comedy	9
13	1920s	Drama	3

10.Find top 3 frequent actor-pairings in top rated movies.



```
-- 10. Find top 3 frequent actor pairings in top rated movies.
SELECT MAX(imdb_rating) AS max_rating,star1,star2,COUNT(*) AS num_movies
FROM movie_details
GROUP BY star1,star2
ORDER BY num_movies DESC LIMIT 3;
```

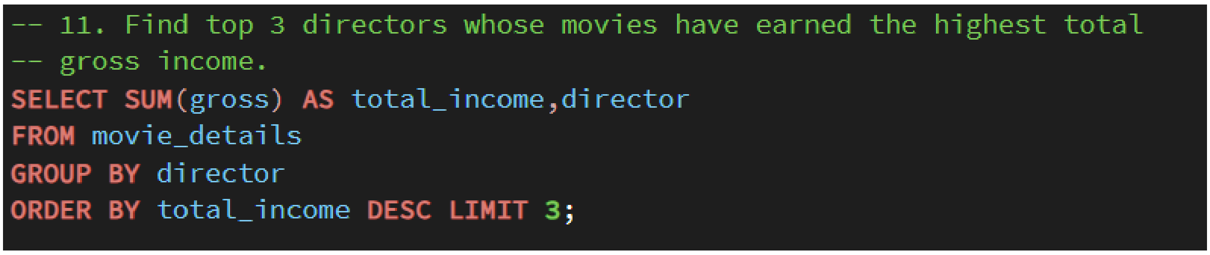
Output:



	max_rating double precision	star1 character varying (80)	star2 character varying (80)	num_movies bigint
1	8.1	Daniel Radcliffe	Emma Watson	5
2	8.3	Tom Hanks	Tim Allen	3
3	8.7	Mark Hamill	Harrison Ford	3

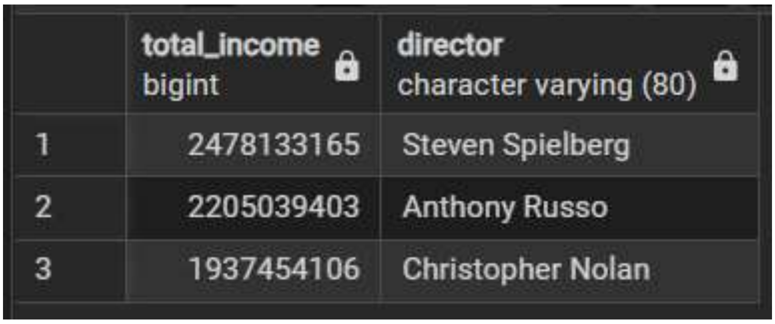
The pairing of **Daniel Radcliffe** and **Emma Watson** appears in the most movies in the top 1000, followed by **Tom Hanks** and **Tim Allen**, and **Mark Hamill** and **Harrison Ford**.

11. Find top 3 directors whose movies have earned the highest total gross income.



```
-- 11. Find top 3 directors whose movies have earned the highest total
-- gross income.
SELECT SUM(gross) AS total_income,director
FROM movie_details
GROUP BY director
ORDER BY total_income DESC LIMIT 3;
```

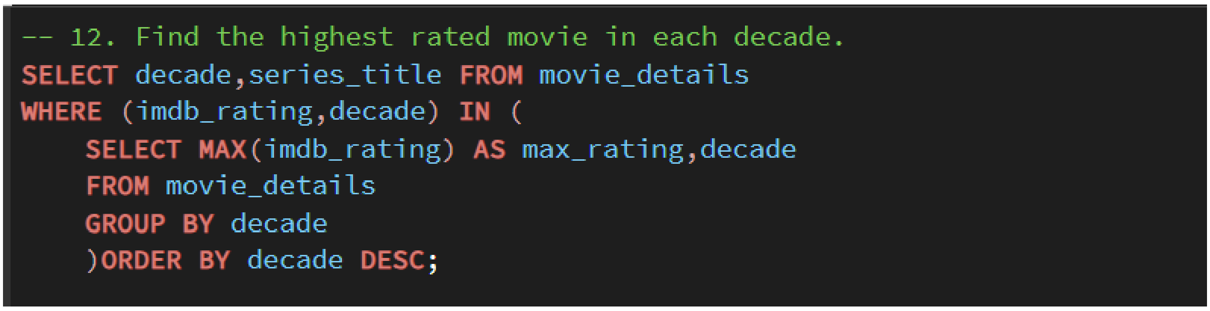
Output:



	total_income bigint	director character varying (80)
1	2478133165	Steven Spielberg
2	2205039403	Anthony Russo
3	1937454106	Christopher Nolan

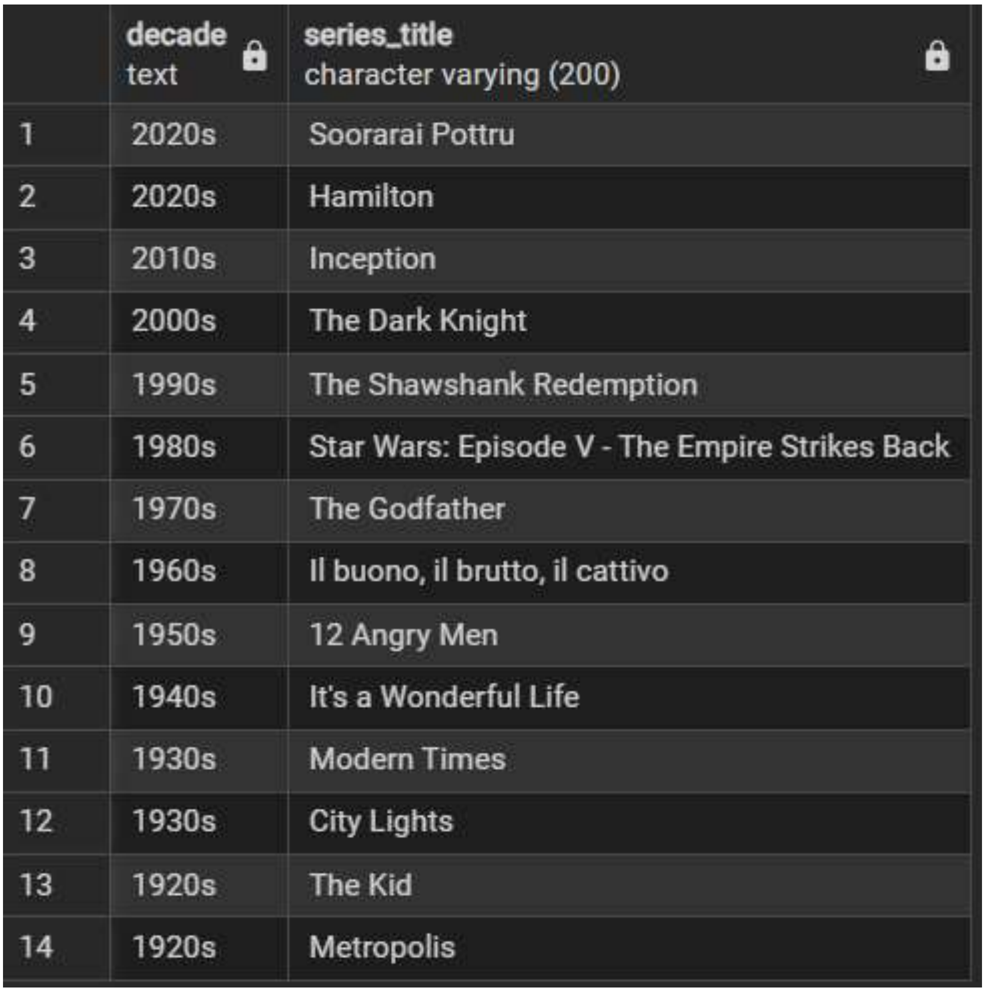
The movies of **Stephen Spielberg, Anthony Russo**, and **Christopher Nolan** have earned the highest gross income within the top 1000.

12.Find the highest rated movie in each decade.



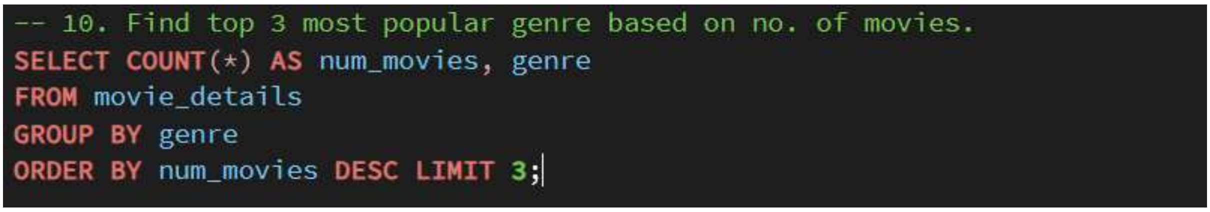
```
-- 12. Find the highest rated movie in each decade.
SELECT decade,series_title FROM movie_details
WHERE (imdb_rating,decade) IN (
    SELECT MAX(imdb_rating) AS max_rating,decade
    FROM movie_details
    GROUP BY decade
    )ORDER BY decade DESC;
```

Output:



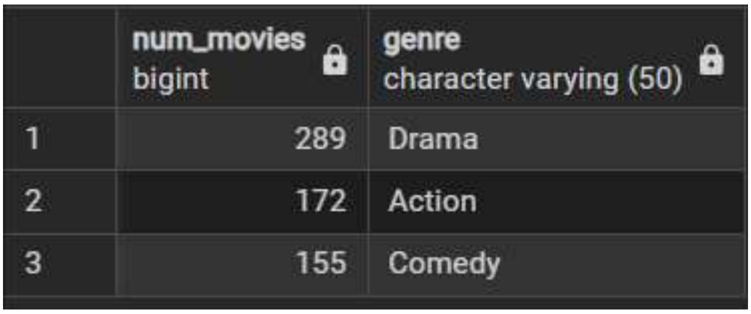
	decade text	series_title character varying (200)
1	2020s	Soorarai Pottru
2	2020s	Hamilton
3	2010s	Inception
4	2000s	The Dark Knight
5	1990s	The Shawshank Redemption
6	1980s	Star Wars: Episode V - The Empire Strikes Back
7	1970s	The Godfather
8	1960s	Il buono, il brutto, il cattivo
9	1950s	12 Angry Men
10	1940s	It's a Wonderful Life
11	1930s	Modern Times
12	1930s	City Lights
13	1920s	The Kid
14	1920s	Metropolis

13.Find top 3 most popular genres (based on no. of movies).



```
-- 10. Find top 3 most popular genre based on no. of movies.
SELECT COUNT(*) AS num_movies, genre
FROM movie_details
GROUP BY genre
ORDER BY num_movies DESC LIMIT 3;
```

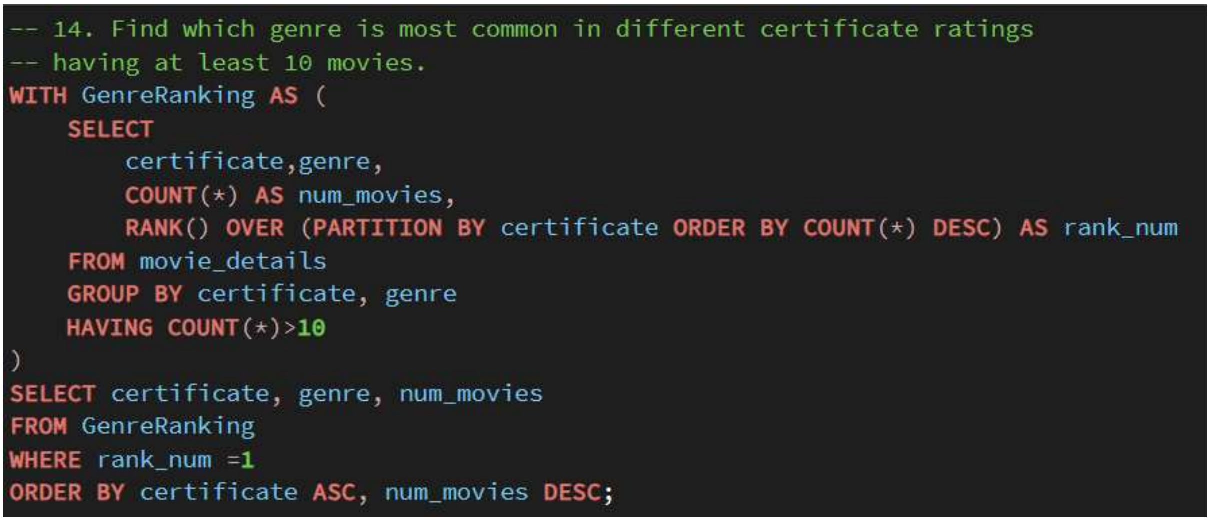
Output:



	num_movies bigint	genre character varying (50)
1	289	Drama
2	172	Action
3	155	Comedy

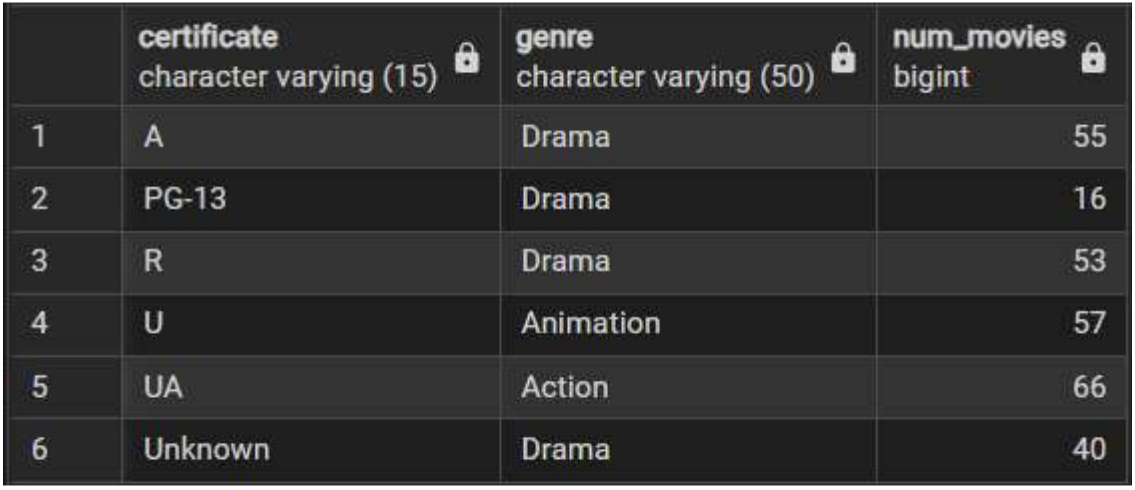
Drama, Action, and **Comedy** are the most popular genres, as determined by the total number of movies.

14. Find which genre is the most common in different certificate ratings having at least 10 movies.



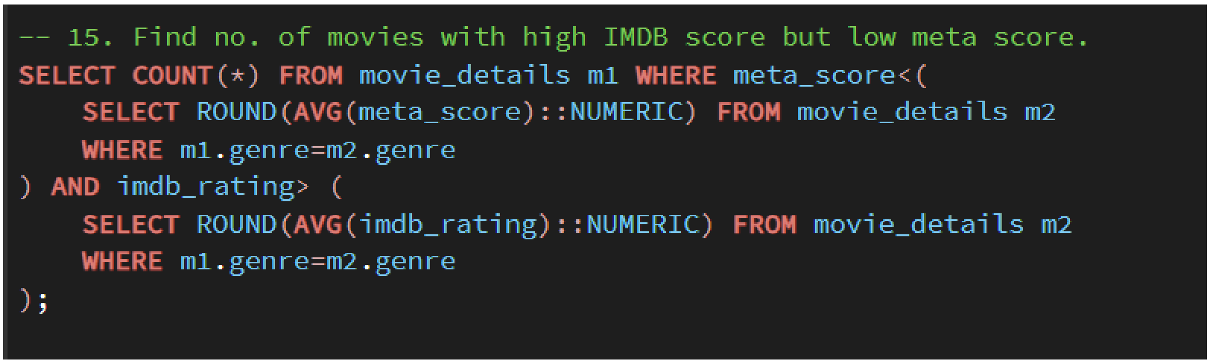
```
-- 14. Find which genre is most common in different certificate ratings
-- having at least 10 movies.
WITH GenreRanking AS (
    SELECT
        certificate,genre,
        COUNT(*) AS num_movies,
        RANK() OVER (PARTITION BY certificate ORDER BY COUNT(*) DESC) AS rank_num
    FROM movie_details
    GROUP BY certificate, genre
    HAVING COUNT(*)>10
)
SELECT certificate, genre, num_movies
FROM GenreRanking
WHERE rank_num =1
ORDER BY certificate ASC, num_movies DESC;
```

Output:



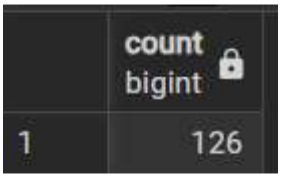
	certificate character varying (15)	genre character varying (50)	num_movies bigint
1	A	Drama	55
2	PG-13	Drama	16
3	R	Drama	53
4	U	Animation	57
5	UA	Action	66
6	Unknown	Drama	40

15.Find no. of movies with high IMDB score but low meta score.



```sql
-- 15. Find no. of movies with high IMDB score but low meta score.
SELECT COUNT(*) FROM movie_details m1 WHERE meta_score<(
    SELECT ROUND(AVG(meta_score)::NUMERIC) FROM movie_details m2
    WHERE m1.genre=m2.genre
) AND imdb_rating> (
    SELECT ROUND(AVG(imdb_rating)::NUMERIC) FROM movie_details m2
    WHERE m1.genre=m2.genre
);
```

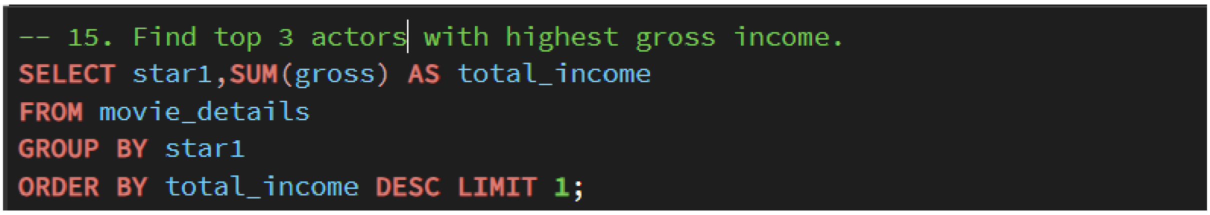
Output:



	count bigint
1	126

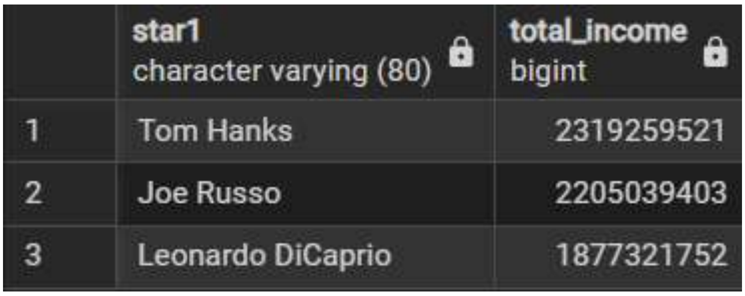
Despite having high IMDB scores, **126 of the 999 movies** have low Meta scores.

16.Find top 3 actors with highest gross income.



```
-- 15. Find top 3 actors with highest gross income.
SELECT star1,SUM(gross) AS total_income
FROM movie_details
GROUP BY star1
ORDER BY total_income DESC LIMIT 1;
```

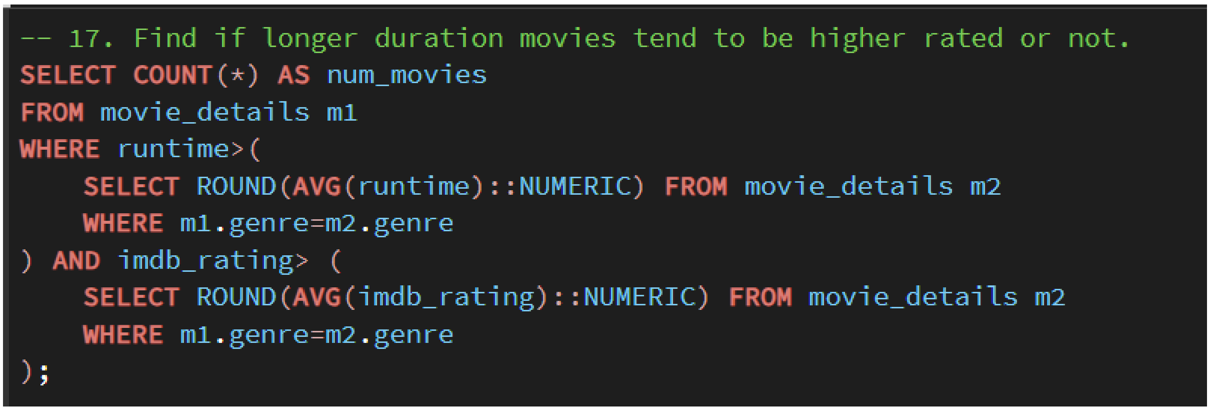
Output:



	star1 character varying (80)	total_income bigint
1	Tom Hanks	2319259521
2	Joe Russo	2205039403
3	Leonardo DiCaprio	1877321752

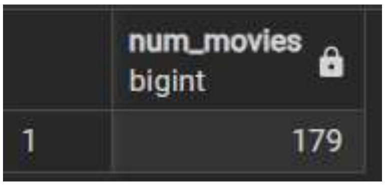
The films of **Tom Hanks** have achieved the highest income, followed by those of **Joe Russo** and **Leonardo DiCaprio**.

17.Find if longer duration movies tend to be higher rated or not.



```
-- 17. Find if longer duration movies tend to be higher rated or not.
SELECT COUNT(*) AS num_movies
FROM movie_details m1
WHERE runtime>(
    SELECT ROUND(AVG(runtime)::NUMERIC) FROM movie_details m2
    WHERE m1.genre=m2.genre
) AND imdb_rating> (
    SELECT ROUND(AVG(imdb_rating)::NUMERIC) FROM movie_details m2
    WHERE m1.genre=m2.genre
);
```

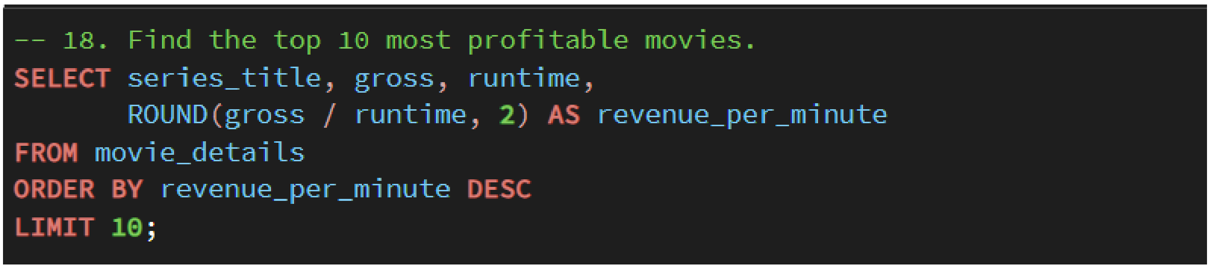
Output:



	num_movies bigint
1	179

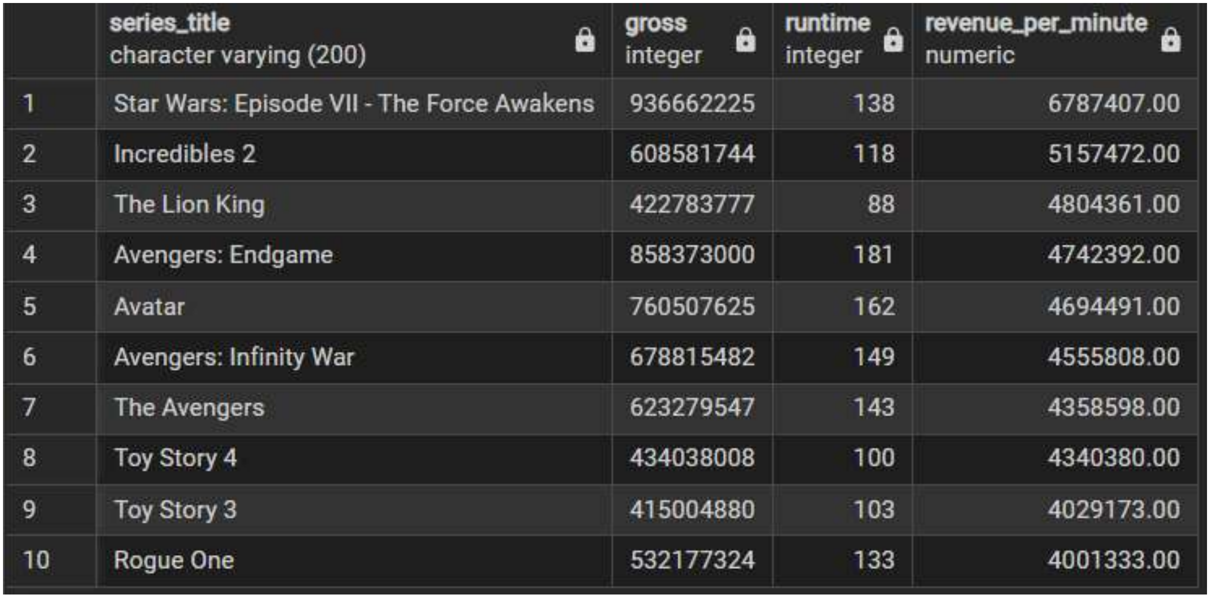
Only **179 of the 999 movies** have durations and ratings that are higher than the average for their respective genres.

18.Find the top 10 most profitable movies. (Based on gross and runtime)



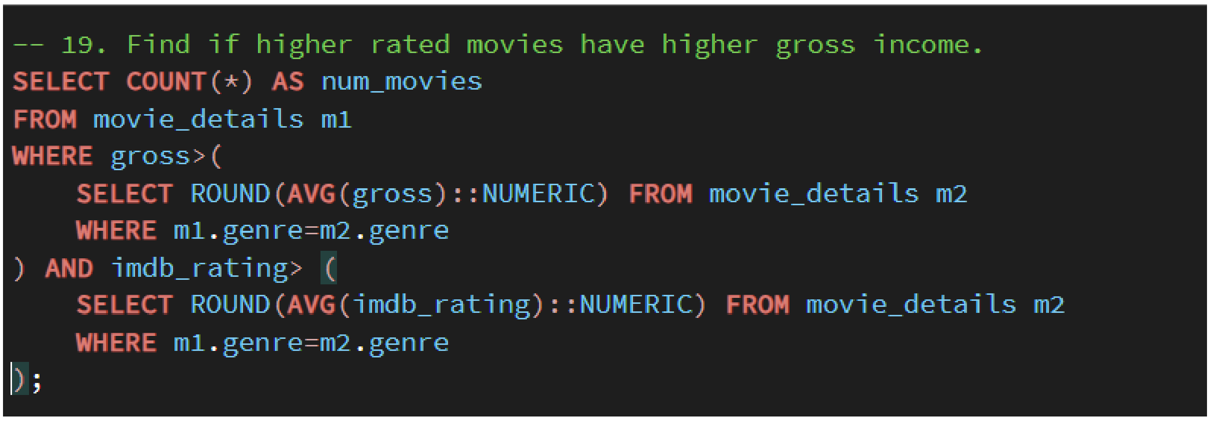
```sql
-- 18. Find the top 10 most profitable movies.
SELECT series_title, gross, runtime,
       ROUND(gross / runtime, 2) AS revenue_per_minute
FROM movie_details
ORDER BY revenue_per_minute DESC
LIMIT 10;
```

Output:



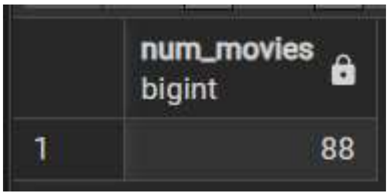
	series_title character varying (200)	gross integer	runtime integer	revenue_per_minute numeric
1	Star Wars: Episode VII - The Force Awakens	936662225	138	6787407.00
2	Incredibles 2	608581744	118	5157472.00
3	The Lion King	422783777	88	4804361.00
4	Avengers: Endgame	858373000	181	4742392.00
5	Avatar	760507625	162	4694491.00
6	Avengers: Infinity War	678815482	149	4555808.00
7	The Avengers	623279547	143	4358598.00
8	Toy Story 4	434038008	100	4340380.00
9	Toy Story 3	415004880	103	4029173.00
10	Rogue One	532177324	133	4001333.00

19.Find if higher rated movies have higher gross income.



```
-- 19. Find if higher rated movies have higher gross income.
SELECT COUNT(*) AS num_movies
FROM movie_details m1
WHERE gross>(
    SELECT ROUND(AVG(gross)::NUMERIC) FROM movie_details m2
    WHERE m1.genre=m2.genre
) AND imdb_rating> (
    SELECT ROUND(AVG(imdb_rating)::NUMERIC) FROM movie_details m2
    WHERE m1.genre=m2.genre
);
```

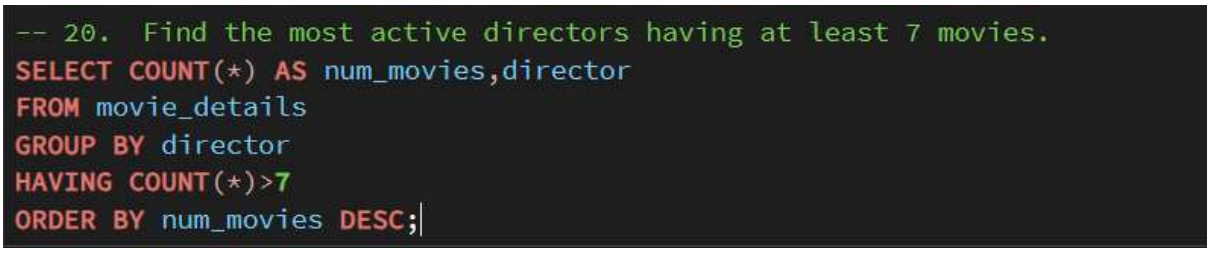
Output:



	num_movies bigint
1	88

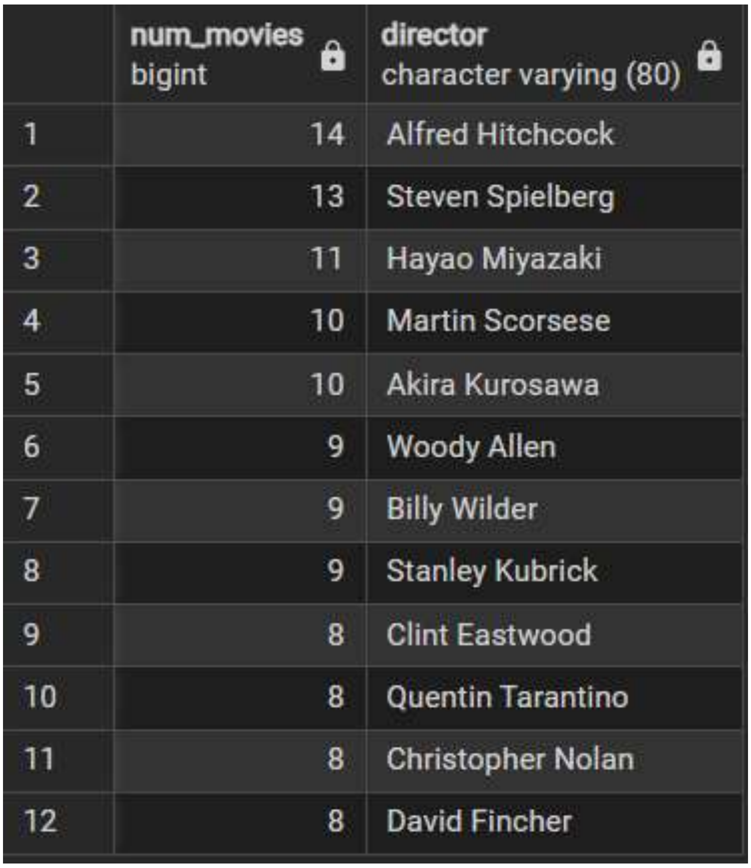
88 of the 999 movies are both high-rated and high-grossing.

20.Find the most active directors having at least 7 movies.



```sql
-- 20.  Find the most active directors having at least 7 movies.
SELECT COUNT(*) AS num_movies,director
FROM movie_details
GROUP BY director
HAVING COUNT(*)>7
ORDER BY num_movies DESC;
```

Output:



	num_movies bigint	director character varying (80)
1	14	Alfred Hitchcock
2	13	Steven Spielberg
3	11	Hayao Miyazaki
4	10	Martin Scorsese
5	10	Akira Kurosawa
6	9	Woody Allen
7	9	Billy Wilder
8	9	Stanley Kubrick
9	8	Clint Eastwood
10	8	Quentin Tarantino
11	8	Christopher Nolan
12	8	David Fincher