5000+ Movies From IMDB (Internet Movie Database)

Предметная область представляет собой информацию о фильмах, их рейтинге, выставленном пользователями сайта IMDB, жанрах фильмов, актерах и режиссерах, странах и проч.

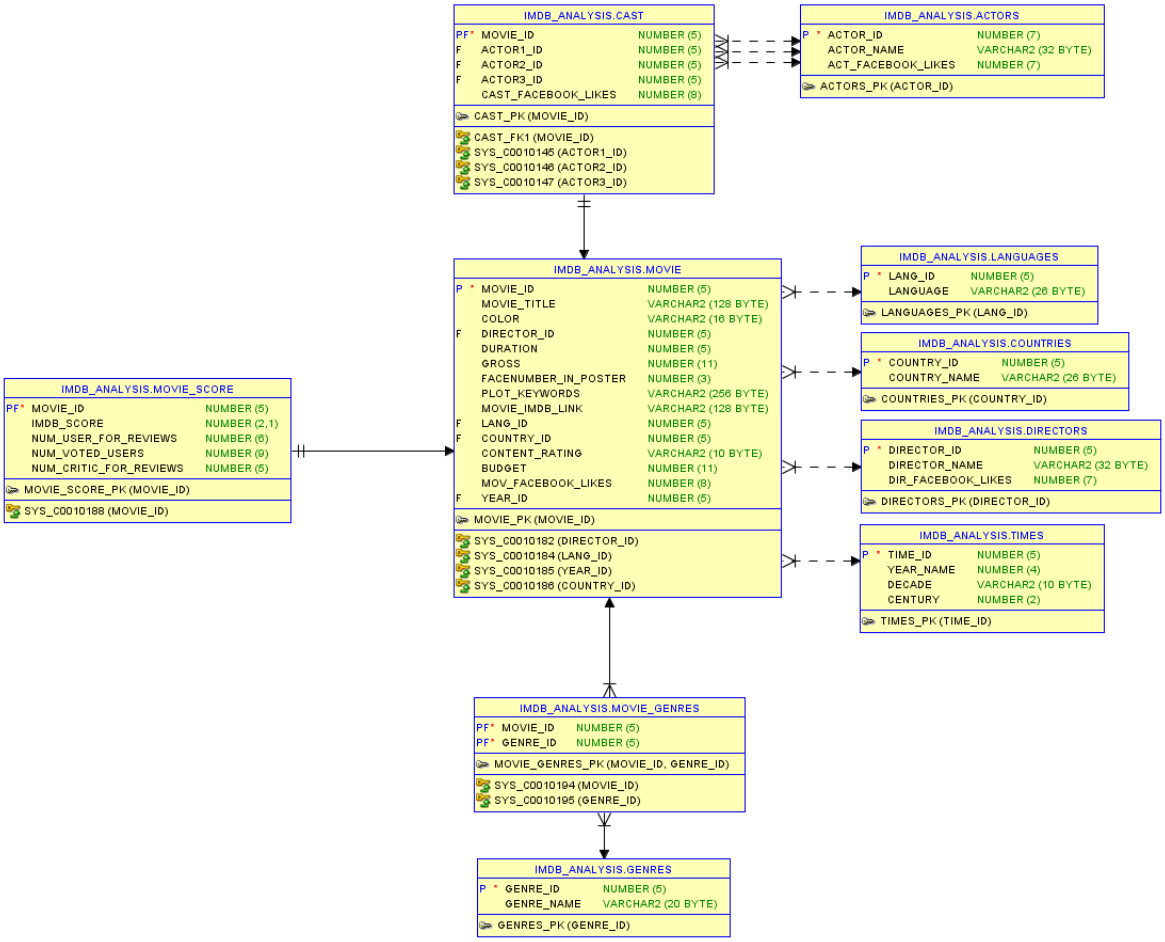


Таблица фактов – Movie_score. Фактом является рейтинг фильма, количество человек, голосовавших за фильм, количество отзывов критиков и посетителей сайта. Таблица содержит внешний ключ на измерение Фильм (movie).

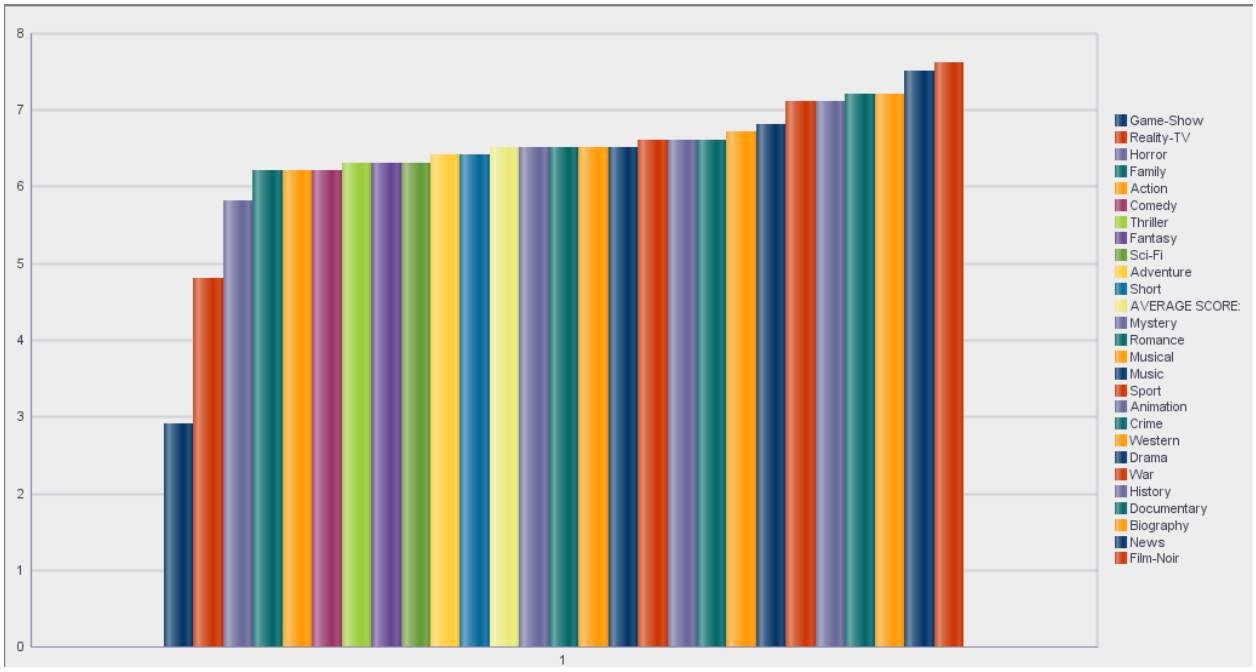
У измерения movie также имеются свои измерения – Genres (жанры, свзяь многие ко многим), Languages, Actors (связь многие ко многим), Director, Times (содержит года выпуска фильмов, десятилетия, века), Countries.

Запросы

1. Средний рейтинг по жанрам с итогом

```
Select decode(GROUPING_ID(genre_name),1,'AVERAGE
SCORE:',genre_name) genre,
round(avg(imdb_score),1) as avg_score
from genres join movie_genres using (genre_id)
```

join movie using (movie_id)
join MOVIE_SCORE using (movie_id)
group by grouping sets ((), genre_name);



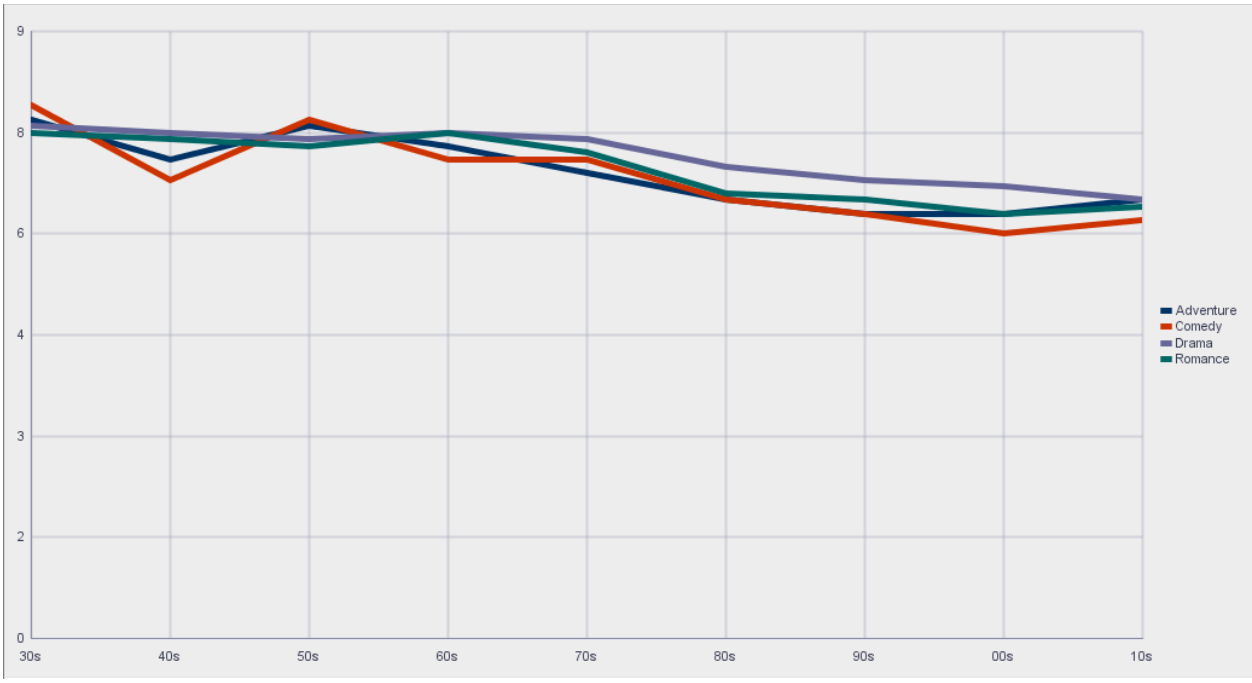
2. Изменение средних оценок некторых жанров по десятилетиям

```
select genre, "30s", "40s", "50s", "60s", "70s", "80s", "90s", "00s", "10s"
from (
Select genre_name genre,
round(avg(imdb_score),1) "30s",
lead(round(avg(imdb_score),1),1) over (partition by genre_name order by century, decade) "40s",
lead(round(avg(imdb_score),1),2) over (partition by genre_name order by century, decade) "50s",
lead(round(avg(imdb_score),1),3) over (partition by genre_name order by century, decade) "60s",
lead(round(avg(imdb_score),1),4) over (partition by genre_name order by century, decade) "70s",
lead(round(avg(imdb_score),1),5) over (partition by genre_name order by century, decade) "80s",
lead(round(avg(imdb_score),1),6) over (partition by genre_name order by century, decade) "90s",
lead(round(avg(imdb_score),1),7) over (partition by genre_name order by century, decade) "00s",
lead(round(avg(imdb_score),1),8) over (partition by genre_name order by century, decade) "10s",
rank() over (partition by genre_name order by century, decade) rang
from genres join movie_genres using (genre_id)
```

```
join movie using (movie_id)
join MOVIE_SCORE using (movie_id)
join times on year_id=time_id
Where genre_name in ('Adventure', 'Comedy', 'Romance', 'Drama') and decade in
('30s','40s','50s','60s','70s','80s','90s','00s','10s')
group by  genre_name,century, decade
)
where rang=1
;
```

3. Изменение средних оценок некторых жанров по десятилетиям (тот же смысл, но друая визуализация)

```
Select genre_name genre,decade,
round(avg(imdb_score),1)
from genres join movie_genres using (genre_id)
join movie using (movie_id)
join MOVIE_SCORE using (movie_id)
join times on year_id=time_id
Where genre_name in ('Adventure', 'Comedy', 'Romance', 'Drama') and decade in
('30s','40s','50s','60s','70s','80s','90s','00s','10s')
group by  genre_name,century, decade
order by 1,century,2;
```



4. Top 5 best actors

```
Select actor_name, score
From (
Select actor_name, round(avg(imdb_score),1) as score,
dense_rank() over (order by round(avg(imdb_score),1) desc) rang
```

```
from actors join cast on actors.actor_id=cast.ACTOR1_ID or
actors.actor_id=cast.ACTOR2_ID or actors.actor_id=cast.ACTOR1_ID
join movie using (movie_id)
join MOVIE_SCORE using (movie_id)
group by actor_name)
where rang<=5;
```

	ACTOR_NAME	SCORE
1	Andrea Martin	9,5
2	Krystyna Janda	9,1
3	Olaf Lubaszenko	9,1
4	T.J. Storm	9,1
5	Jeffrey DeMunn	8,9
6	Luigi Pistilli	8,9
7	Lee J. Cobb	8,9
8	Royce Johnson	8,8
9	Kenny Baker	8,8
10	Aziz Muradillayev	8,7
11	Maria Pia Calzone	8,7
12	Luke Edwards	8,7
13	Art Carney	8,7
14	Abigail Evans	8,7
15	Kimberley Crossman	8,7
16	Stacie Evans	8,7
17	Marcus Chong	8,7
18	Peter Cushing	8,7
19	Elina Abai Kyzy	8,7
20	Fortunato Cerlino	8,7
21	Minoru Chiaki	8,7

5. Top 5 best directors

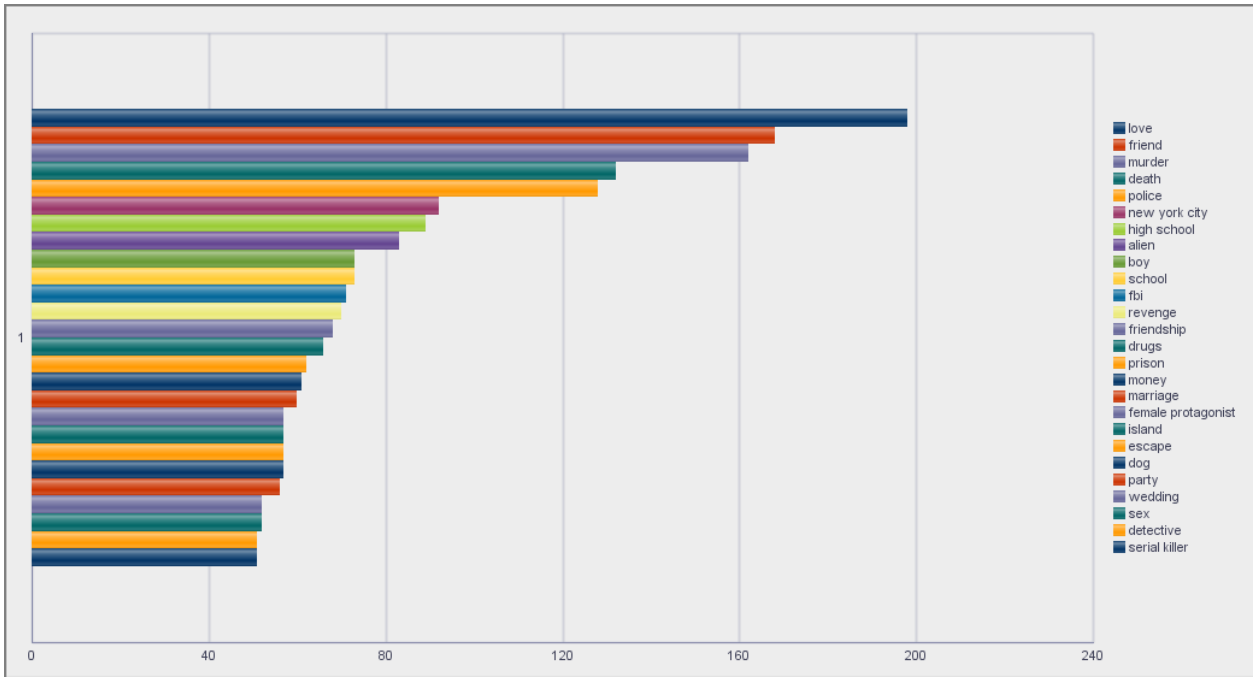
```
Select director_name, score
From (
Select director_name, round(avg(imdb_score),1) as score,
dense_rank() over (order by round(avg(imdb_score),1) desc) rang
from directors
right join movie using (director_id)
join MOVIE_SCORE using (movie_id)
group by director_name)
where rang<=5;
```

	DIRECTOR_NAME	SCORE
1	John Blanchard	9,5
2	Cary Bell	8,7
3	Sadyk Sher-Niyaz	8,7
4	Mitchell Altieri	8,7
5	Charles Chaplin	8,6
6	Mike Mayhall	8,6
7	Sergio Leone	8,5
8	Raja Menon	8,5
9	Majid Majidi	8,5
10	Ron Fricke	8,5
11	Damien Chazelle	8,5
12	Marius A. Markevicius	8,4
13	Asghar Farhadi	8,4
14	Bill Melendez	8,4
15	Jay Oliva	8,4
16	Moustapha Akkad	8,4
17	Christopher Nolan	8,4
18	Robert Mulligan	8,4
19	Catherine Owens	8,4
20	S.S. Rajamouli	8,4
21	Rakeysh Omprakash Mehra	8,4

6. most frequently used keywords in movies

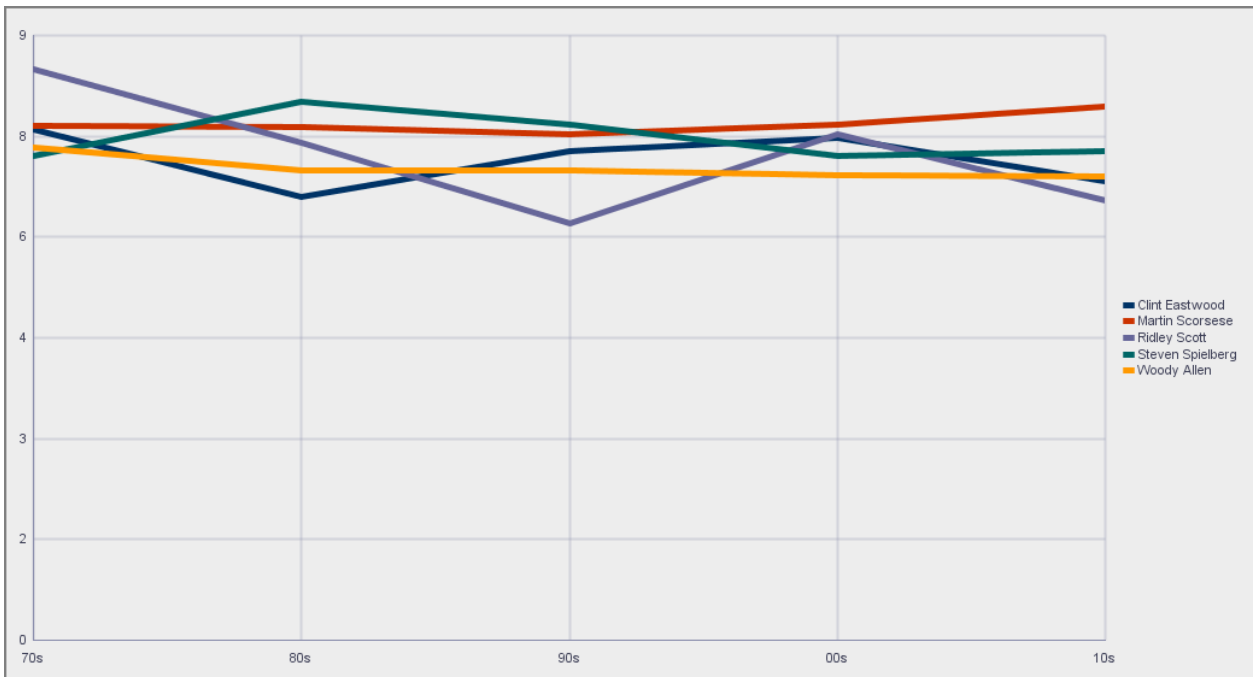
```
Select keywords, amount from (
Select keywords, count(*) amount, dense_rank() over (order by count(*) desc)
rang
from (
Select substr(plot_keywords,1,instr(plot_keywords, '|')-1) as keywords
from movie
union all
select substr(plot_keywords,instrb(plot_keywords, '|',1,1)+1,instrb(plot_keywords,
'|',1,2)-(instrb(plot_keywords, '|',1,1)+1))
from movie
union all
select substr(plot_keywords,instrb(plot_keywords, '|',1,2)+1,instrb(plot_keywords,
'|',1,3)-(instrb(plot_keywords, '|',1,2)+1))
from movie
union all
select substr(plot_keywords,instrb(plot_keywords, '|',1,3)+1,instrb(plot_keywords,
'|',1,4)-(instrb(plot_keywords, '|',1,3)+1))
from movie
union all
select substr(plot_keywords,instrb(plot_keywords, '|',1,4)+1,instrb(plot_keywords,
'|',1,5)-(instrb(plot_keywords, '|',1,4)+1))
```

```
from movie
union all
select substr(plot_keywords,instrb(plot_keywords, '|',1,5)+1,instrb(plot_keywords,
'|',1,6)-(instrb(plot_keywords, '|',1,5)+1))
from movie
union all
select substr(plot_keywords,instrb(plot_keywords, '|',1,6)+1,instrb(plot_keywords,
'|',1,7)-(instrb(plot_keywords, '|',1,6)+1))
from movie
union all
select  substr(plot_keywords,instrb(plot_keywords, '|',1,7)+1,instrb(plot_keywords,
'|',1,8)-(instrb(plot_keywords, '|',1,7)+1))
from movie)
where keywords is not null
group by keywords)
where rang<=20
;
```



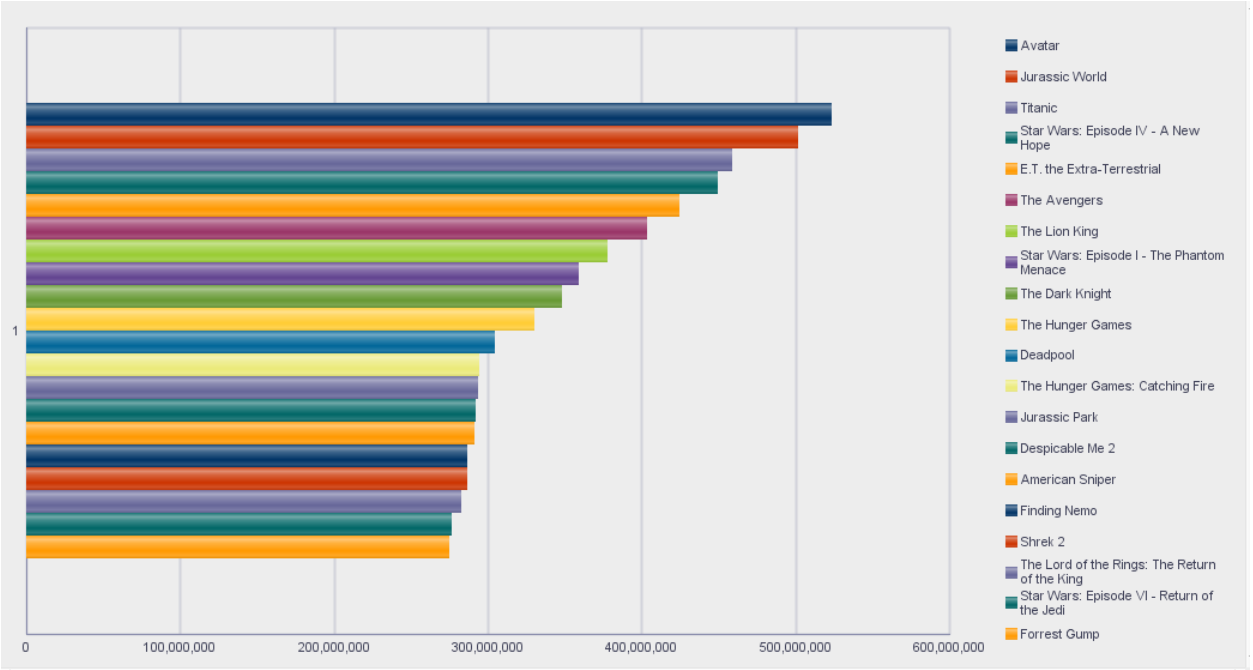

```
--most productive directors and their score in time
Select decade, director_name,  avg(imdb_score) as score
From directors
join movie using (director_id)
join times on year_id=time_id
join MOVIE_SCORE using (movie_id)
Where director_name IN (Select director_name
From (
Select director_name, count(movie_id) as score
From directors
```

```
join movie using (director_id)
group by director_name
order by 2 desc
)
where rownum<=5)
Group by director_name, decade, century
order by 2, century, 1
;
```



7. movies with highest net profit

```
Select movie_title, net
From (
Select distinct movie_title, gross-budget as net, dense_rank() over (order by gross-budget desc) rang
from movie
where gross is not null and budget is not null
order by net desc
)
where rang<=20;
```


1
0
100,000,000
200,000,000
300,000,000
400,000,000
500,000,000
600,000,000
Avatar
Jurassic World
Titanic
Star Wars: Episode IV - A New Hope
E.T. the Extra-Terrestrial
The Avengers
The Lion King
Star Wars: Episode I - The Phantom Menace
The Dark Knight
The Hunger Games
Deadpool
The Hunger Games: Catching Fire
Jurassic Park
Despicable Me 2
American Sniper
Finding Nemo
Shrek 2
The Lord of the Rings: The Return of the King
Star Wars: Episode VI - Return of the Jedi
Forrest Gump