

Proyecto Final

“Análisis de Plataformas de Streaming”

Armando Gómez

Introducción a Bases de Datos

5/12/2020

1.- ¿Cuáles son los ratings de las 5 películas de mayor puntuación en IMDb?

```
select * from rating order by IMDb desc limit 5;
```

```
1 • select * from rating order by IMDb desc limit 5;|
2
3
4
5
6
7
```

100% 49:1			
Result Grid			
Filter Rows: Search Export: Fe			
Id_movCritics	IMDb	RottenTomatoes	
6567	9.3	0.78	
7221	9.3	1.33	
7427	9.3	2.5	
5111	9.3	2	
6838	9.3	0.44	
rating 13			

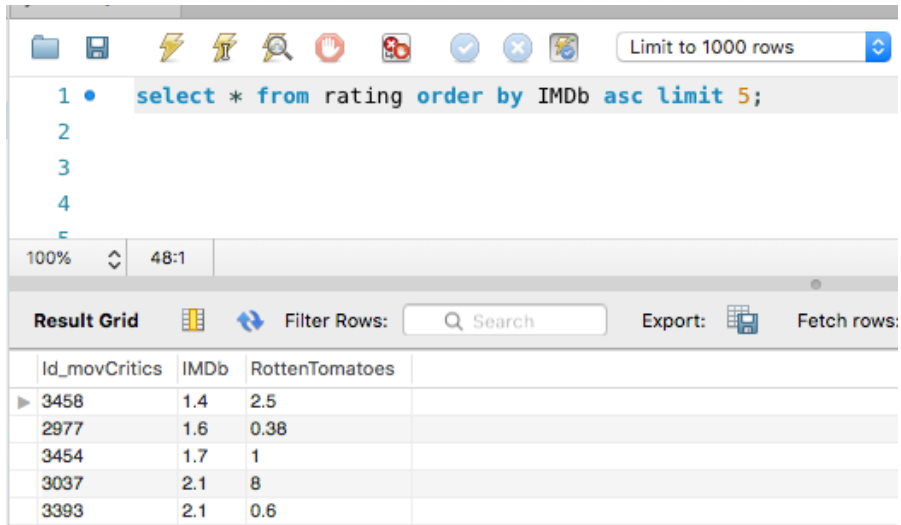
Project// {_id:0, RottenTomatoes: 0}

Sort // { IMDb : -1}

FILTER	
PROJECT { _id:0, RottenTomatoes: 0 }	
SORT { IMDb : -1 }	
COLLATION	
VIEW	
id_Movies: 7427	IMDb: 9.3
id_Movies: 5111	IMDb: 9.3
id_Movies: 7221	IMDb: 9.3
id_Movies: 6567	IMDb: 9.3
id_Movies: 6838	IMDb: 9.3

2.- ¿Cuáles son las 5 películas de menor puntuación en IMDb?

```
select * from rating order by IMDb asc limit 5;
```

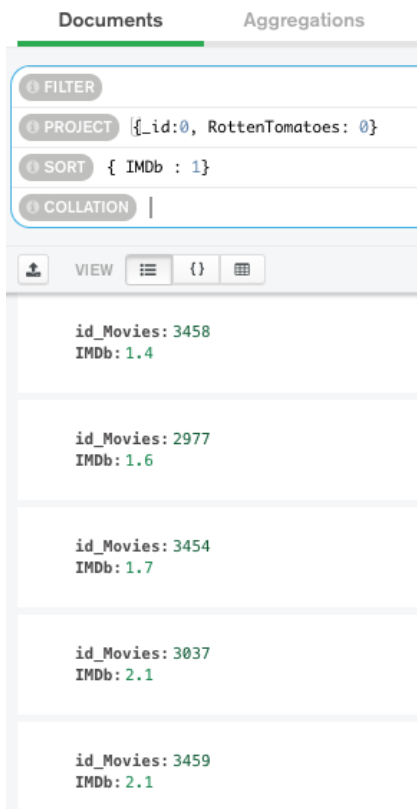


The screenshot shows a database query interface with a toolbar at the top containing icons for file operations, search, and execution. A text area contains the SQL query: `select * from rating order by IMDb asc limit 5;`. Below the query, a 'Result Grid' section displays the results in a table format. The table has three columns: 'Id_movCritics', 'IMDb', and 'RottenTomatoes'. The results are sorted by IMDb rating in ascending order, showing the top 5 movies with the lowest ratings.

Id_movCritics	IMDb	RottenTomatoes
3458	1.4	2.5
2977	1.6	0.38
3454	1.7	1
3037	2.1	8
3393	2.1	0.6

Project// {_id:0, RottenTomatoes: 0}

Sort // { IMDb : 1}



The screenshot shows a MongoDB query interface with a 'Documents' tab selected. The query is defined with a 'PROJECT' stage: `{_id:0, RottenTomatoes: 0}` and a 'SORT' stage: `{ IMDb : 1}`. The results are displayed in a list of documents, each showing the movie ID and its IMDb rating.

id_Movies	IMDb
3458	1.4
2977	1.6
3454	1.7
3037	2.1
3459	2.1

3.- ¿Cuáles son los 5 directores de mayor edad?

```
select Directors, Edad from directors order by Edad desc limit 5;
```

3 • `select Directors, Edad from directors order by Edad desc limit 5;`

4

5

100% 66:3

Result Grid Filter Rows: Search Export: Fetch rows:

Directors	Edad
Michael Angelo	73
Beth Toni Kruvant	73
Edward Dmytryk	73
Jo Daniel Tikhomiroff	73
Mack V. Wright	73

```
Project // {_id:0, Names:1, Edad:1}
Sort// { Edad : -1}
```

FILTER

PROJECT {_id:0, Names:1, Edad:1}

SORT { Edad : -1}

COLLATION

VIEW { }

Names: "Igor Kovalyov,Norton Virgien"	Edad: 73
Names: "Nathan Morlando"	Edad: 73
Names: "Tinge Krishnan"	Edad: 73
Names: "Amol Palekar"	Edad: 73
Names: "Jay Chapman"	Edad: 73

4.- ¿Cuántas películas tiene cada plataforma?

```
select count(Netflix) as Peliculas from platforms where Netflix = 1
Union
select count(Hulu) from platforms where Hulu = 1
union
select count(PrimeVideo) from platforms where PrimeVideo = 1;
```

```

5 • select count(Netflix) as Peliculas from platforms where Netflix = 1
6   Union
7   select count(Hulu) from platforms where Hulu = 1
8   union
9   select count(PrimeVideo) from platforms where PrimeVideo = 1;
10

```

Peliculas
3376
869
5981

```
Filter // {Netflix : {$eq: 1}}
Filter // {Hulu : {$eq: 1}}
Filter // {PrimeVideo : {$eq: 1}}
```

<div> <div>FILTER {Netflix : {\$eq: 1}}</div> <div>OPTIONS</div> <div>FIND</div> </div>	
PROJECT	{_id:0}
SORT	
COLLATION	
<div> <div>MAXTIME</div> <div>5000</div> </div>	
<div> <div>SKIP</div> <div>0</div> </div>	
<div> <div>LIMIT</div> <div>0</div> </div>	
<div> <div>VIEW</div> <div> <div></div> <div></div> <div></div> </div> </div>	
<div>Displaying documents 1 - 20 of 3376</div>	
id_Movies: 1	Netflix: 1
Hulu: 0	PrimeVideo: 0
id_Movies: 2	Netflix: 1
Hulu: 0	PrimeVideo: 0

The top screenshot shows the MongoDB Compass interface with the following query:

```

    FILTER: {Hulu : {$eq: 1}}
    PROJECT: {_id:0}
    SORT:
    COLLATION:
    MAXTIMEMS: 5000
    SKIP: 0
    LIMIT: 0
  
```

The results show two documents:

```

    id_Movies: 68
    Netflix: 1
    Hulu: 1
    PrimeVideo: 0

    id_Movies: 98
    Netflix: 1
    Hulu: 1
    PrimeVideo: 0
  
```

The bottom screenshot shows the same interface with a different query:

```

    FILTER: {PrimeVideo : {$eq: 1}}
    PROJECT: {_id:0}
    SORT:
    COLLATION:
    MAXTIMEMS: 5000
    SKIP: 0
    LIMIT: 0
  
```

The results show two documents:

```

    id_Movies: 5
    Netflix: 1
    Hulu: 0
    PrimeVideo: 1

    id_Movies: 7
    Netflix: 1
    Hulu: 0
    PrimeVideo: 1
  
```

5.- ¿Cuántas películas del genero acción hay en total en las 3 plataformas?

```
select count(Genres) as PeliculasDeAccion from movies where Genres Like '%Action%';
```

The screenshot shows the SQL query results in a database client. The query is:

```
select count(Genres) as PeliculasDeAccion from movies where Genres Like '%Action%';
```

The results show a single row with the value 1397.

PeliculasDeAccion
1397

```
Filter// { Genres: /Action/i }
```

Proyecto_Final.Movies

DOCUMENTS 9.6k TOTAL SIZE 1.9MB AVG. SIZE 202B INDEXES 1

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { Genres: /Action/i } OPTIONS

PROJECT

SORT MAXTIMES 5000

COLLATION SKIP 0 LIMIT 0

ADD DATA VIEW {}

Displaying documents 1 - 20 of 1416

```
> {
  "_id": ObjectId("5fc9c953ddfa9e03c6865c6e"),
  "id_Movies": 1,
  "Title": "Inception",
  "Year": 2010,
  "Age": "13+",
  "id_director": 1,
  "Genres": "Action,Adventure,Sci-Fi,Thriller",
  "Country": "United States,United Kingdom",
  "Language": "English,Japanese,French",
  "Runtime": 148
}
```

6.- ¿Cuáles son las 10 mayores puntuaciones en Tomato Roten y en IMDb?

```
select RottenTomatoes from rating
order by RottenTomatoes desc limit 10;
```

```
14 • select RottenTomatoes from rating
15 order by RottenTomatoes desc limit 10;
16
```

100% 1:6

Result Grid Filter Rows: Search Export:

RottenTomatoes
1
1
1
1
0.99
0.99
0.99
rating 5

```
select IMDb from rating
order by IMDb desc limit 10;
```

17 • `select IMDb from rating`
 18 `order by IMDb desc limit 10;`
 19

IMDb
9.3
9.3
9.3
9.3
9.3
9.1
9.1

rating 6

```
filter={}
project={
  '_id': 0,
  'IMDb': 1
}
sort=list({
  'IMDb': -1
})
```

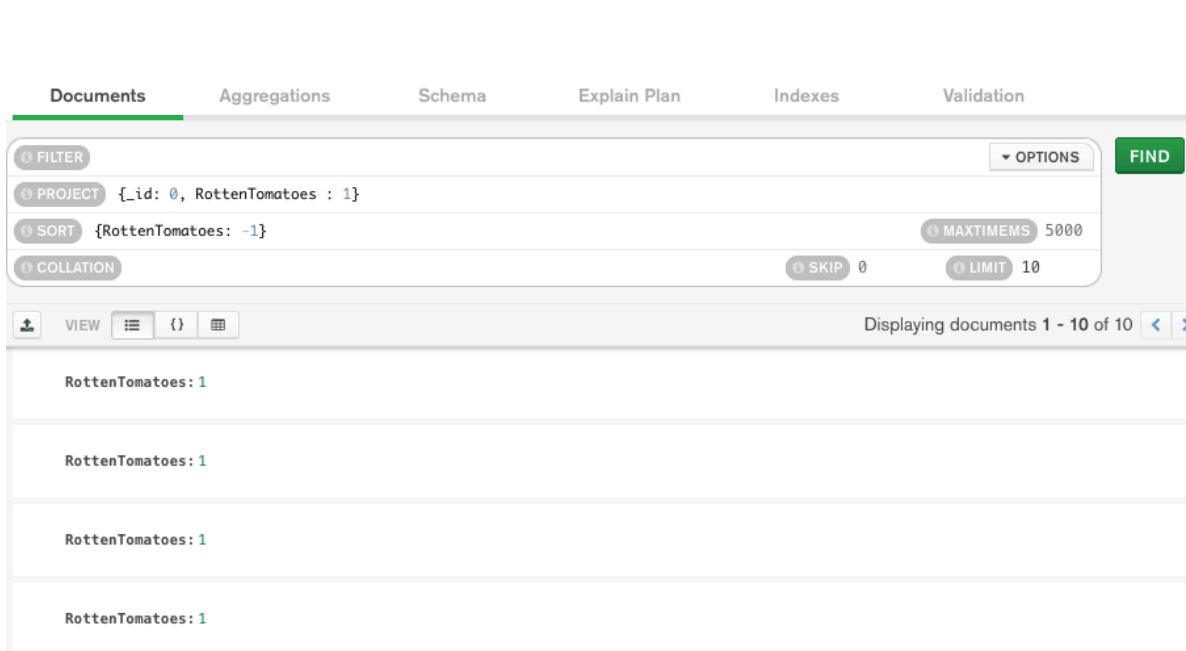
PROJECT	SORT	MAXTIMES	COLLATION	SKIP	LIMIT
{_id: 0, IMDb: 1}	{IMDb: -1}	5000		0	10

VIEW [] [] []

Displaying documents 1 - 10 of 10

IMDb: 9.3
IMDb: 9.3
IMDb: 9.3
IMDb: 9.3



```
project={
  '_id': 0,
  'RottenTomatoes': 1
}
sort=list({
  'RottenTomatoes': -1
}.items())
limit=10
```







7.- ¿Qué directores tienen las 10 películas más recientes?

```
select directors.Directors Nombre, movies.Title Título, movies.Year Año
from movies
join directors on
movies.ID_Director = directors.id_Director
order by año desc;
```

```
26 • select directors.Directors Nombre, movies.Title Título, movies.Year Año
27 from movies
28 join directors on
29 movies.ID_Director = directors.id_Director
30 order by año desc;
31
```

100%  1:24

Result Grid   Filter Rows: Export:  Fetch rows: 

Nombre	Título	Año	
Jagdeep Sidhu	Sufna	2020	
Anthony Mandler	Happiness Continues	2020	
Srividya Basawa	Madha	2020	
Alex Diaz	Alex Fernandez: The Best Comedian in the World	2020	
Navjot Gulati	Jai Mummy Di	2020	
Stanley Moore, Alex Woo	Go! Go! Cory Carson: The Chrissy	2020	
Mikel Rueda	Fronteras	2020	
Jos Humphrey, Kenny Park	Carmen Sandiego: To Steal or Not to Steal	2020	
Malik Nejer	Masameer: The Movie	2020	
Susan Johnson	To All the Boys I've Loved Before	2020	
Don Michael Paul	Bulletproof 2	2020	

Result 31

```
[{$sort: {
  Year: -1
}}, {$lookup: {
  from: "Directors",
  localField: "id_director",
  foreignField: "id_Director",
  as: "Mov_Directors"
}}]
```

\$lookup

Output after [\\$lookup](#) stage ⓘ (Sample of 20 documents)

```

1  /**
2   * from: The target collection.
3   * localField: The local join field.
4   * foreignField: The target join field.
5   * as: The name for the results.
6   * pipeline: The pipeline to run on the joined collection.
7   * let: Optional variables to use in the pipeline.
8   */
9  {
10   from: "Directors",
11   localField: "id_director",
12   foreignField: "id_Director",
13   as: "Mov_Directors"
14 }

```

Genres: "Comedy,Drama,Romance"

Country: "United States"

Language: "English"

Runtime: 99

▼ Mov_Directors: Array

▼ 0: Object

_id: ObjectId("5f9c91eddfa9e03c6863058")

id_Director: 53

Names: "Susan Johnson"

FechaNacimiento: 1972-11-12T06:00:00.000+00:00

PaisNacimiento: "United States"

Edad: 48

Genres: "Action,Thriller"

Country: "United States"

Language: "English,Hindi,Bengali"

Runtime: 116

▼ Mov_Directors: Array

▼ 0: Object

_id: ObjectId("5f9c91eddfa9e03c6863059")

id_Director: 93

Names: "Sam Hargrave"

FechaNacimiento: 1985-11-12T06:00:00.000+00:00

PaisNacimiento: "Beliz"

Edad: 35

8.- ¿En qué plataforma se encuentran las 10 películas con mayor puntuación en IMDb?

```
select rating.IMDb RatingIMDb, platforms.Netflix Netflix, platforms.Hulu Hulu,
platforms.PrimeVideo Prime
from rating
join platforms on rating.Id_movCritics = platforms.ID_MOV
order by IMDb desc limit 10;
```

```
34 • select rating.IMDb RatingIMDb, platforms.Netflix Netflix, platforms.Hulu Hulu, platforms.PrimeVideo Prime
35 from rating
36 join platforms on rating.Id_movCritics = platforms.ID_MOV
37 order by IMDb desc limit 10;
```

100% 20:30

Result Grid Filter Rows: Search Export: Fetch rows:

RatingIMDb	Netflix	Hulu	Prime
9.3	0	0	1
9.3	0	0	1
9.3	0	0	1
9.3	0	0	1
9.3	0	0	1
9.1	0	0	1
9.1	0	0	1
9.1	1	0	0
9	1	0	0
9	1	0	0

Result 35 Read C

```
[{$sort: {
  IMDb: -1
}}, {$lookup: {
  from: "Platforms",
  localField: "id_Movies",
  foreignField: "id_Movies",
  as: "Platf_Rating"
}}, {$addFields: {
  PlataformaObj:{$arrayElemAt: ["$Platf_Rating", 0]}
}}, {$addFields: {
  Netflix: "$PlataformaObj.Netflix",
  Hulu: "$PlataformaObj.Hulu",
  Prime: "$PlataformaObj.PrimeVideo"
}}, {$project: {
  _id: 0,
  Platf_Rating:0,
  PlataformaObj:0,
  RottenTomatoes:0
}}, {$limit: 10}]
```

||| \$limit Output after \$limit stage (Sample of 10 documents)

```
1 //**
2 * Provide the number of documents to limit.
3 */
4 10
```

id_Movies: 7427

IMDb: 9.3

Netflix: 0

Hulu: 0

Prime: 1

id_Movies: 5111

IMDb: 9.3

Netflix: 0

Hulu: 0

Prime: 1

9.- ¿En qué plataforma se encuentran las 10 películas con mayor puntuación en Rotten Tomatoes?

```
select rating.RottenTomatoes RatingRotten, platforms.Netflix Netflix,
platforms.Hulu Hulu, platforms.PrimeVideo Prime
from rating
join platforms on rating.Id_movCritics = platforms.ID_MOV
order by RottenTomatoes desc limit 10;
```

```
39 • select rating.RottenTomatoes RatingRotten, platforms.Netflix Netflix, platforms.Hulu Hulu, platforms.PrimeVideo Prime
40 from rating
41 join platforms on rating.Id_movCritics = platforms.ID_MOV
42 order by RottenTomatoes desc limit 10;
43
44
45
```

RatingRotten	Netflix	Hulu	Prime
10	1	0	0
10	0	0	1
10	1	0	0
10	1	0	0
10	1	0	0
10	0	0	1
10	1	0	0
10	1	0	0
10	1	0	0
10	1	0	0

Proyecto_Final.Rating

DOCUMENTS 9.6k TOTAL SIZE 706.6KB AVG. SIZE 75B INDEXES 1 TOTAL SIZE 100.0KB A 1C

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE SAMPLE MODE AUTO PREVIEW

COLLATION { locale: 'simple' }

```

1 from: The target collection.
2 * localField: The local join field.
3 * foreignField: The target join field.
4 * as: The name for the results.
5 * pipeline: The pipeline to run on the joined collection.
6 * let: Optional variables to use in the pipeline.
7 */
8 {
9   $project: {
10     from: "Platforms",
11     localField: "id_Movies",
12     foreignField: "id_Movies",
13     as: "Platf_Rating"
14   }
15 }

```

Output after \$project stage (Sample of 20 documents)

```

1 /**
2  * specifications: The fields to
3  * include or exclude.
4  */
5 {
6   _id: 0, RottenTomatoes: 1, Platf_Rating: 1
7 }

```

RottenTomatoes: 1

Platf_Rating: Array

0: Object

```

_id: ObjectId("5fc9c976ddfa9e03c686a3b5")
id_Movies: 8920
IMDb: 8.8
RottenTomatoes: 1
Platf_Rating: Array

```

RottenTomatoes: 1

Platf_Rating: Array

0: Object

```

_id: ObjectId("5fc9c976ddfa9e03c686a3b5")
id_Movies: 8920
IMDb: 8.8
RottenTomatoes: 1
Platf_Rating: Array

```

```
[{$sort: {
  RottenTomatoes: -1
}}, {$lookup: {
  from: "Platforms",
  localField: "id_Movies",
  foreignField: "id_Movies",
  as: "Platf_Rating"
}}, {$addFields: {
  PlataformaObj:{$arrayElemAt: ["$Platf_Rating", 0]}
}}, {$addFields: {
  Netflix: "$PlataformaObj.Netflix",
  Hulu: "$PlataformaObj.Hulu",
  Prime: "$PlataformaObj.PrimeVideo"
}}, {$project: {
  _id: 0,
  Platf_Rating:0,
  PlataformaObj:0,
  IMDb:0
}}, {$limit: 10}]
```

Proyecto_Final.Rating

DOCUMENTS 9.6k TOTAL SIZE 706.6KB AVG. SIZE 75B INDEXES 1 TOTAL SIZE 132.0KB AV

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE SAMPLE MODE AUTO PREVIEW

\$project Output after \$project stage (Sample of 20 documents)

```
1 /**
2  * specifications: The fields to
3  * include or exclude.
4  */
5 {
6   _id: 0,
7   Platf_Rating:0,
8   PlataformaObj:0,
9   IMDb:0
10 }
```

id_Movies: 8920
RottenTomatoes: 1
Netflix: 0
Hulu: 0
Prime: 1

id_Movies: 9234
RottenTomatoes: 1
Netflix: 0
Hulu: 0
Prime: 1

\$limit Output after \$limit stage (Sample of 10 documents)

```
1 /**
2  * Provide the number of documents to limit.
3  */
4 10
```

id_Movies: 8920
RottenTomatoes: 1
Netflix: 0
Hulu: 0
Prime: 1

id_Movies: 9234
RottenTomatoes: 1
Netflix: 0
Hulu: 0
Prime: 1

10.- ¿En qué plataforma se encuentran las 10 películas más recientes?

```
select movies.Year, platforms.Hulu Hulu, platforms.Netflix Netflix,
platforms.PrimeVideo Prime
from movies
join platforms
on movies.ID = platforms.ID_MOV
order by movies.Year DESC limit 10;
```

```
44 • select movies.Year, platforms.Hulu Hulu, platforms.Netflix Netflix, platforms.PrimeVideo Prime
45   from movies
46   join platforms
47   on movies.ID = platforms.ID_MOV
48   order by movies.Year DESC limit 10;
49
50
```

100%

39:42

Result Grid

Filter Rows:

Export:

Fetch rows:

	Year	Hulu	Netflix	Prime	
▶	2020	0	1	0	
	2020	0	1	0	
	2020	0	1	0	
	2020	0	1	0	
	2020	0	1	0	
	2020	0	1	0	
	2020	0	1	0	
	2020	0	1	0	
	2020	0	1	0	
	2020	0	1	0	

```
[{$sort: {
  Year: -1
}}, {$lookup: {
  from: "Platforms",
  localField: 'id_Movies',
  foreignField: 'id_Movies',
  as: 'string'
}}, {$addFields: {
  PlatObj: {$arrayElemAt: ["$string", 0] }
}}, {$addFields: {
  Netflix: "$PlatObj.Netflix",
  Hulu: "$PlatObj.Hulu",
  Prime: "$PlatObj.PrimeVideo"
}}, {$project: {
  _id:0,
  string:0,
  Age:0,
  PlatObj:0,
  Language:0,
  Country:0,
  Runtime:0,
  id_director:0
}}, {$limit: 10}, {$count: 'Netflix'}]
```

Proyecto_Final.Movies

DOCUMENTS 9.6k TOTAL SIZE 1.9MB AVG. SIZE 202B INDEXES 1 TOTAL SIZE 100.0KB AVG. SIZE 100.0

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE SAMPLE MODE AUTO PREVIEW

Output after \$limit stage (Sample of 10 documents)

```

1 //**
2 * Provide the number of documents to limit.
3 */
4 10

```

id_Movies: 159
Title: "Miss Americana"
Year: 2020
Genres: "Documentary"
Netflix: 1
Hulu: 0
Prime: 0

id_Movies: 113
Title: "Extraction"
Year: 2020
Genres: "Action,Thriller"
Netflix: 1
Hulu: 0
Prime: 0

Output after \$count stage (Sample of 1 document)

```

1 //**
2 * Provide the field name for the count.
3 */
4 'Netflix'

```

Netflix: 10

11.- Crea una vista que contenga ID y titulo de película, nombre del director y el promedio de calificación entre IMDb y Rotten Tomatoes.

```

create view PromRating as
select movies.ID ID_Movie,
       movies.Title Titulo,
       directors.Directors Nombre,
       ((IMDb + RottenTomatoes*10)/2) Rating_Promedio
from directors
join movies on directors.id_Director = movies.ID_Director
join rating on rating.Id_movCritics = movies.ID;

```

87 • select * from promrating;

100% 1:84

Result Grid Filter Rows: Search Export: Fetch rows:

ID_Movie	Titulo	Nombre	Rating_Promedio
1	Inception	Christopher Nolan	9.25
2	The Matrix	Lana Wachowski,Lilly Wachowski	9.149999999999999
84	The Matrix Reloaded	Lana Wachowski,Lilly Wachowski	6.449999999999999
3	Avengers: Infinity War	Anthony Russo,Joe Russo	8.95
4	Back to the Future	Robert Zemeckis	9.1
63	Back to the Future Part III	Robert Zemeckis	7.85
54	Back to the Future Part II	Robert Zemeckis	8.55
88	The Polar Express	Robert Zemeckis	6.85

--Creo la Vista del lookup entre Movies y Rating donde calculo el Rating Promedio, con esto solo me falta agregar con otro lookup el nombre del director de la tabla Directors.

```
[{$lookup: {
  from: 'Movies',
  localField: 'id_Movies',
  foreignField: 'id_Movies',
  as: 'Movies'
}}, {$addFields: {
  MovObj: {$arrayElemAt:["$Movies", 0] }
}}, {$addFields: {
  Movies_id:"$MovObj.id_Movies",
  Title: "$MovObj.Title",
  id_director:"$MovObj.id_director"
}}, {$project: {
  Title: 1,
  Movies_id: 1,
  id_director: 1,
  Suma:

      {$sum:
    [{$multiply:
      ["$RottenTomatoes", 10]},"$IMDb"]

    }
  }
}}, {$project: {
  Title: 1,
  Movies_id: 1,
  id_director: 1,
  Promedio : {
    $divide : [ "$Suma", 2 ]
  }
}}]
```

--Creo una segunda vista entre la primer vista y Directors para agregar el nombre del director.

```
[{$lookup: {
  from: 'Directors',
  localField: 'id_director',
  foreignField: 'id_Director',
  as: 'DirectorInfo'
}}, {$addFields: {
  DirectorObj: {$arrayElemAt:["$DirectorInfo", 0] }
}}, {$addFields: {
  Name: "$DirectorObj.Names"
}}, {$project: {
  Movies_id: 1,
  Title: 1,
  Name: 1,
  Promedio: 1,

  }}}]
```


Proyecto_Final.RatingProm (view on: Proyecto_Final.Rating)

Documents Aggregations Schema Explain Plan Indexes

FILTER

VIEW

```

_id: ObjectId("5fcad99a4aa73901e5c4978a")
Movies_id: 2926
Title: "A New Capitalism"
id_director: 2290
Promedio: 6.699999999999999

_id: ObjectId("5fcad99a4aa73901e5c49791")
Movies_id: 4009
Title: "Afro Samurai: Resurrection"
id_director: 3031
Promedio: 7.4

_id: ObjectId("5fcad99a4aa73901e5c497b6")
Movies_id: 1906
Title: "A Cinderella Story: Christmas Wish"
id_director: 1536
Promedio: 5.4

```

12.- ¿Qué plataforma tiene más películas del director con mayor puntuación en sus películas?

```

select Nombre, Rating_Promedio from promrating order by Rating_Promedio DESC;
select * from directors where Directors like 'Miguel Gau%';
select * from movies where ID_Director = 5267;
select * from platforms where ID_MOV = 7221;

```

```

81 • select Nombre, avg(Rating_Promedio) Promedio
82   from promrating
83   group by Nombre order by Promedio desc;
84

```

100% 25:74

Result Grid Filter Rows: Search Export:

Nombre	Promedio
► Miguel Gaudencio	9.9
Fen Tian	9.7
Frank Marshall	9.6
Paul Kakert	9.6
Tolga Ornek	9.6
Don Carroll, Noriko Carroll	9.55
Bryan Buckley	9.55

```

90 • select Nombre, Rating_Promedio from promrating order by Rating_Promedio DESC;
91 • select * from directors where Directors like 'Miguel Gau%';
92 • select * from movies where ID_Director = 5267;
93 • select * from platforms where ID_MOV = 7221;
94
95

```

100% 1:88

Result Grid Filter Rows: Search Export:

ID_MOV	Netflix	Hulu	PrimeVideo
► 7221	0	0	1

13.- ¿En que plataforma están las 20 películas mejor evaluadas en promedio entre IMDb y Rotten Tomatoes?

```

create view Best20 as
select promrating.Rating_Promedio Rating, promrating.Titulo Titulo,
       platforms.Netflix Netflix,
       platforms.Hulu Hulu,
       platforms.PrimeVideo Prime
from promrating
join platforms on promrating.ID_Movie = platforms.ID_MOV
order by Rating desc limit 20;

```

100% 30:104

Result Grid Filter Rows: Search Export:

	Rating	Titulo	Netflix	Hulu	Prime
▶	9.9	Down, But Not Out!	0	0	1
	9.75	The Jungle School	1	0	0
	9.7	Love on a Leash	0	0	1
	9.6	Cop Watchers	1	0	0
	9.6	Escape from Firebase Kate	0	0	1
	9.6	You're Everything To Me	1	0	0
	9.6	The Dark Knight	0	1	0
	9.6	Alvin	0	1	0

```
select sum(Netflix) Netflix, sum(Prime) Prime, sum(Hulu) Hulu from best20;
```

106 • `select sum(Netflix) Netflix, sum(Prime) Prime, sum(Hulu) Hulu from best20;`

100% 5:100


Result Grid Filter Rows: Search Export:



	Netflix	Prime	Hulu
▶	5	13	2


14.- ¿Qué plataforma tiene mayor cantidad de cortometrajes?

```
create view shorts as
select movies.ID ID, movies.Title Titulo, platforms.Netflix, platforms.Hulu,
platforms.PrimeVideo
from movies
right join platforms on movies.ID = platforms.ID_MOV
where Genres like '%Short%';
```

```
134 • select * from shorts;
135
```

100%  1:129

Result Grid   Filter Rows:


Export: 




ID	Titulo	Netflix	Hulu	PrimeVideo
▶ 250	The White Helmets	1	0	0
424	Period. End of Sentence.	1	0	0
596	Fire in Paradise	1	0	0
641	Heroin(e)	1	0	0
870	Dream Big: Engineering Our World	1	0	0
922	The Lonely Island Presents: The Unauthorized...	1	0	0
993	National Parks Adventures	1	0	0
1000	Maze Runner: One Hundred Days	1	0	0

charts 11

```
select sum(Netflix) ShortsOnNetflix,
       sum(PrimeVideo) ShortsOnPrime,
       sum(Hulu) ShortsOnHulu
from shorts;
```

```
129 • select sum(Netflix) ShortsOnNetflix,
130         sum(PrimeVideo) ShortsOnPrime,
131         sum(Hulu) ShortsOnHulu
132 from shorts;
133
134
135
```

100%  29:127

Result Grid   Filter Rows: Export: 

ShortsOnNetflix	ShortsOnPrime	ShortsOnHulu	
▶ 78	88	7	

15.- ¿Qué plataforma tiene más contenido en minutos?

```
create view Runtime as
select movies.Runtime, movies.Title, platforms.Hulu Hulu, platforms.Netflix,
platforms.PrimeVideo
from movies
join platforms on movies.ID = platforms.ID_MOV;
```

```
134 • create view Runtime as
135 select movies.Runtime, movies.Title, platforms.Hulu Hulu, platforms.Netflix, platforms.PrimeVideo
136 from movies
137 join platforms on movies.ID = platforms.ID_MOV;
```

100%	3:131	
Result Grid		
Filter Rows: <input type="text" value="Search"/>		
Export:		
Fetch rows:		
Runtime	Title	Hulu Netflix PrimeVideo
148	Inception	0 1 0
136	The Matrix	0 1 0
149	Avengers: Infinity War	0 1 0
116	Back to the Future	0 1 0
161	The Good, the Bad and the Ugly	0 1 1
117	Spider-Man: Into the Spider-Verse	0 1 0
150	The Pianist	0 1 1
125	Die Hard: With a Vengeance	0 1 0

```
select sum(Hulu) Hulu, sum(Netflix) Netflix, sum(PrimeVideo) Prime from runtime;
```

```
144 • select sum(Hulu) Hulu, sum(Netflix) Netflix, sum(PrimeVideo) Prime from runtime;
```

100%	1:146	
Result Grid		
Filter Rows: <input type="text" value="Search"/>		
Export:		
Hulu	Netflix	Prime
863	3263	5939