

# TP 03

Kopenkin Dmitrii

## A - Movies

1. Quelle est la note moyenne de tout les films ?

```
use('tp3');
db.movies.aggregate([
  {
    $group: {
      _id: null,
      average: { $avg: '$imdb.rating' },
    },
  },
]);
```

A screenshot of a MongoDB aggregation result displayed in a dark-themed terminal. The result is a single document in an array, represented as a JSON object. The object has two fields: "\_id" with a value of null, and "average" with a value of 6.6934040649367255. The text is color-coded: brackets are yellow, the opening brace is purple, strings are green, null is red, and the average value is orange.

```
[
  {
    "_id": null,
    "average": 6.6934040649367255
  }
]
```

2. À l'aide d'une d'agrégation, donner le nombre de film réalisés par année

```
use('tp3');
db.movies.aggregate([
  {
    $group: {
      _id: '$year',
      count: { $sum: 1 },
    },
  },
]);
```

```

25  ✓ [
24  ✓ {
23    "_id": "1997è",
22    "count": 2
21  },
20  ✓ {
19    "_id": 1981,
18    "count": 179
17  },
16  ✓ {
15    "_id": 1945,
14    "count": 39
13  },
12  > { ...
9    },
8  > { ...
5    },
4  > { ...

```

3. À l'aide d'une pipeline d'agrégation, et de la question 2, donnez les 3 années les plus productive en terme de réalisation de films.

```

use('tp3');
db.movies.aggregate([
  {
    $group: {
      _id: '$year',
      count: { $sum: 1 },
    },
  },
  {
    $sort: {
      count: -1,
    },
  },
  {
    $limit: 3,
  },
]);

```

```
[
  {
    "_id": 2013,
    "count": 1220
  },
  {
    "_id": 2014,
    "count": 1147
  },
  {
    "_id": 2012,
    "count": 1109
  }
]
```

4. Quel est le titre du film, selon imdb, le mieux noté ?

```
use('tp3');
db.movies
  .find({
    'imdb.rating': {
      $ne: '',
    },
  })
  .sort({
    'imdb.rating': -1,
  })
  .limit(1);
```

```
[
  {
    "_id": {
      "$oid": "573a139ff29313caabd003c4"
    },
    "plot": "The story of Easy Company of the US",
    "genres": [
      "Action",
      "Drama",
      "History"
    ],
    "runtime": 705,
    "cast": [
      "Scott Grimes",
      "Matthew Leitch",
      "Damian Lewis",
      "Ron Livingston"
    ],
    "num_mflix_comments": 0,
    "poster": "https://m.media-amazon.com/images/
    ... "title": "Band of Brothers",
    "fullplot": "This is the story of \"E\" Easy
    "languages": [
```

5. Combien de films ont exactement 2 genres ? Combien ont au moins 3 genre ?

```
use('tp3');
db.movies
  .find({
    genres: { $size: 2 },
  })
  .size();
db.movies
  .find({
    genres: { $exists: true },
    $where: 'this.genres.length >= 3',
  })
  .size();
```

1 7366

1 9490

6. Combien de genres un film possède t'il au maximum ?

```

use('tp3');
db.movies.aggregate([
  {
    $sort: {
      'genres.size': 1,
    },
  },
  {
    $limit: 1,
  },
  {
    $addFields: {
      genres_amount: {
        $size: '$genres',
      },
    },
  },
  {
    $project: {
      genres_amount: '$genres_amount',
    },
  },
]);

```

```

[
  {
    "_id": {
      "$oid": "573a1390f29313caabcd4323"
    },
    "genres_amount": 3
  }
]

```

7. Sur l'ensemble des films de la collection, combien de langue différentes trouves-t-on ?

```

use('tp3');
db.movies.distinct('languages').length;

```

```

1    234

```

8. Combien de films sont disponible en langue des signes ? Partial Text Search : Regex ou /search/ (^\*)

```

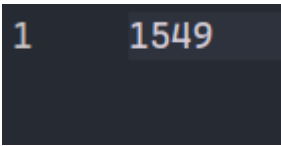
use('tp3');
db.movies.find({ languages: /sign language/i }).size();

```

1 60

9. Donnez le nombre de film sortis après le 19 Mai 2014

```
use('tp3');
db.movies
  .find({
    released: {
      $gt: ISODate('2014-05-19'),
    },
  })
  .size();
```

1 1549

10. Sur les films sortis après le 19 Mai 2014 (Question 8), donnez le nombre d'entre eux qui d'une durée minimum de 2h30

```
use('tp3');
db.movies
  .find({
    released: {
      $gt: ISODate('2014-05-19'),
    },
    runtime: {
      $gt: 150,
    },
  })
  .size();
```

1 24

## B - Movies & Comments

Dans cette partie nous allons mettre en relation la collection movies et comments.

1. Créez une pipeline d'agrégation ajoutant aux documents de la collection movies les commentaires associés dans la collection comments

```
use('tp3');
db.movies.aggregate([
  {
    $lookup: {
      from: 'comments',
      localField: '_id',
```

```

    foreignField: 'movie_id',
    as: 'comments',
  },
},
]);

```

```

},
"lastupdated": "2015-08-29 00:27:45.437000000",
"year": 1912,
"imdb": { ...
},
"countries": [ ...
],
"type": "movie",
"tomatoes": { ...
},
"comments": [
  {
    "_id": {
      "$oid": "5a9427648b0beebeb69579e7"
    },
    "name": "Mercedes Tyler",
    "email": "mercedes_tyler@fakegmail.com",
    "movie_id": {
      "$oid": "573a1390f29313caabcd4323"
    },
    "text": "Eius veritatis vero facilis quaerat fuga temporibus",
    "date": {
      "$date": "2002-08-18T04:56:07Z"
    }
  }
]
},
{

```

2. Donnez le nombre de film ayant au moins 100 commentaires.

```

use('tp3');
db.movies.aggregate([
  {
    $lookup: {
      from: 'comments',
      localField: '_id',
      foreignField: 'movie_id',
      as: 'comments',
    },
  },
],
{

```

```
    $unwind: {
      path: '$comments',
    },
  },
  {
    $group: {
      _id: '$_id',
      count: {
        $sum: 1,
      },
      comments: {
        $push: '$comments',
      },
    },
  },
  {
    $match: {
      count: {
        $gte: 100,
      },
    },
  },
]);
```



```
[
{
  "_id": {
    "$oid": "573a1399f29313caabcec614"
  },
  "count": 134,
  "comments": [
    {
      "_id": {
        "$oid": "5a9427648b0beebeb695e6cc"
      },
      "name": "Anthony Thompson",
      "email": "anthony_thompson@fakegmail.com",
      "movie_id": {
        "$oid": "573a1399f29313caabcec614"
      },
      "text": "Voluptatibus facilis quod veritatis minus quam iu
      "date": {
        "$date": "1982-09-09T20:55:05Z"
      }
    },
    {
      "_id": {
        "$oid": "5a9427648b0beebeb695e6cd"
      },
      "name": "Anthony Smith",
      "email": "anthony_smith@fakegmail.com",
      "movie_id": {
        "$oid": "573a1399f29313caabcec614"
      },
      "text": "Dolore aut aut fuga corporis delectus deleniti. D
      "date": {
        "$date": "1994-06-20T23:38:05Z"
      }
    },
    {
      "_id": {
```

- en fonction de l'environnement, cette requête peut occasionner un timeout. Si pas de résultat, expliquez la requête.

## C - Pipelines d'Agrégation Complexes

1. Créez une agrégation permettant de grouper les films par années et genres dans une nouvelle collection nommée `movies_genres_by_year`. Note : `$unwind` pour exploser un tableau

```
use('tp3');
db.movies.aggregate([
  {
    $group: {
      _id: {
        year: '$year',
        genres: '$genres',
      },
      count: {
        $sum: 1,
      },
    },
  },
  {
    $unwind: '$_id.genres',
  },
  {
    $project: {
      _id: 0,
      year: '$_id.year',
      genres: '$_id.genres',
      count: '$count',
    },
  },
  {
    $out: 'movies_genres_by_year',
  },
]);
```

```
db.getCollection('movies_genres_by_year').find({})
```

movies\_genres\_by\_year 0.002 sec.

	_id	year	genres	count
1	ObjectId("61f7a6f294dd9a09ef680611")	1979	Sci-Fi	1
2	ObjectId("61f7a6f294dd9a09ef680612")	2003	Drama	2
3	ObjectId("61f7a6f294dd9a09ef680613")	2003	Horror	2
4	ObjectId("61f7a6f294dd9a09ef680614")	2003	Sci-Fi	2
5	ObjectId("61f7a6f294dd9a09ef680615")	2003	Adventure	2
6	ObjectId("61f7a6f294dd9a09ef680616")	2003	Animation	2
7	ObjectId("61f7a6f294dd9a09ef680617")	2003	Family	2
8	ObjectId("61f7a6f294dd9a09ef680618")	1982	Action	2
9	ObjectId("61f7a6f294dd9a09ef680619")	1982	Crime	2
10	ObjectId("61f7a6f294dd9a09ef68061a")	1982	Drama	2
11	ObjectId("61f7a6f294dd9a09ef68061b")	2013	Action	5
12	ObjectId("61f7a6f294dd9a09ef68061c")	2013	Adventure	5
13	ObjectId("61f7a6f294dd9a09ef68061d")	2013	Fantasy	5
14	ObjectId("61f7a6f294dd9a09ef68061e")	1947	Drama	1
15	ObjectId("61f7a6f294dd9a09ef68061f")	1947	Romance	1
16	ObjectId("61f7a6f294dd9a09ef680620")	2009	Adventure	1
17	ObjectId("61f7a6f294dd9a09ef680621")	2009	Fantasy	1
18	ObjectId("61f7a6f294dd9a09ef680622")	2009	Mystery	1
19	ObjectId("61f7a6f294dd9a09ef680623")	2011	History	1
20	ObjectId("61f7a6f294dd9a09ef680624")	2012	Drama	5
21	ObjectId("61f7a6f294dd9a09ef680625")	2012	Fantasy	5
22	ObjectId("61f7a6f294dd9a09ef680626")	2011	Action	3
23	ObjectId("61f7a6f294dd9a09ef680627")	2011	Sci-Fi	3
24	ObjectId("61f7a6f294dd9a09ef680628")	2011	Thriller	3
25	ObjectId("61f7a6f294dd9a09ef680629")	1968	Comedy	2

2. Créez une pipeline d'agrégation pour créer une collection de documents avec les informations suivantes :














Directeur	Nombre de films	Victoires	Nominations
String	Int	Int	Int

```
use('tp3');
db.movies.aggregate([
  {
    $unwind: '$directors',
  },
  {
    $group: {
      _id: '$directors',
      count: {
        $sum: 1,
      },
      victories: {
        $sum: '$awards.wins',
      },
      nominations: {
        $sum: '$awards.nominations',
      },
    },
  },
])
```

```

    },
  },
  {
    $project: {
      _id: 0,
      director: '$_id',
      count: '$count',
      victories: '$victories',
      nominations: '$nominations',
    },
  },
  ],
]);

```

	director	count	victories	nominations
8	 Hermine Huntgeburth	 4.0	 6	 6
9	 Jay DiPietro	 1.0	 0	 1
10	 Lajos Koltai	 2.0	 5	 8
11	 Robert Young	 2.0	 0	 5
12	 Daniel Burman	 5.0	 26	 41
13	 Phil Karlson	 3.0	 2	 3
14	 Sagar Ballary	 1.0	 1	 4
15	 Gez Medinger	 1.0	 1	 0
16	 Hans Steinbichler	 1.0	 5	 7
17	 Flemming Quist Møller	 2.0	 2	 1
18	 Markku Pèlènen	 7.0	 29	 11
19	 Olatunde Osunsanmi	 2.0	 0	 2
20	 Frederic Golchan	 1.0	 1	 0
21	 Carlos Bolado	 2.0	 14	 5

3. Créez une pipeline d'agrégation permettant de récupérer la note moyenne d'IMDB par pays de production.

```

use('tp3');
db.movies.aggregate([
  {
    $unwind: '$countries',
  },
  {
    $group: {
      _id: '$countries',
      average: {
        $avg: '$imdb.rating',
      },
    },
  },
]);

```

```
    },
  ]);
```

```
[
  {
    "_id": "Liberia",
    "average": 7.533333333333334
  },
  {
    "_id": "Switzerland",
    "average": 6.8900000000000001
  },
  {
    "_id": "Chad",
    "average": 6.6400000000000001
  },
  {
    "_id": "Montenegro",
    "average": 7.05
  },
  {
    "_id": "United Arab Emirates",
    "average": 6.8533333333333335
  },
  {
    "_id": "Kyrgyzstan",
    "average": 7.325
  },
  {
    "_id": "Colombia",
    "average": 7.048484848484848
  },
  {
    "_id": "Guatemala",
    "average": 7.15
  },
  {
    "_id": "Angola",
    "average": 6.666666666666667
  },
]
```

4. Créez une pipeline d'agrégation permettant de récupérer le nombre de film disponibles par langues.

```
use('tp3');
db.movies.aggregate([
  {
```

```
    $unwind: '$languages',  
  },  
  {  
    $group: {  
      _id: '$languages',  
      count: {  
        $sum: 1,  
      },  
    },  
  },  
},  
]);
```

```
[  
  {  
    "_id": "Finnish",  
    "count": 283  
  },  
  {  
    "_id": "Ryukyuan",  
    "count": 1  
  },  
  {  
    "_id": "Quenya",  
    "count": 1  
  },  
  {  
    "_id": "Shanghainese",  
    "count": 20  
  },  
  {  
    "_id": "Bulgarian",  
    "count": 26  
  },  
  {  
    "_id": "Russian",  
    "count": 919  
  },  
  {  
    "_id": "Frisian",  
    "count": 2  
  },  
  {  
    "_id": "Syriac",  
    "count": 3  
  },  
  {  
    "_id": "Arabic",  
    "count": 365  
  }  
]
```