

MongoDB Aggregation Pipelining Project

User Schema:

```
_id: ObjectId('66d42b02251a7de92d86f56f')
index : 13
name : "Sharon Grimes"
isActive : true
registered : 2017-12-22T01:04:16.000+00:00
age : 28
gender : "female"
eyeColor : "brown"
favoriteFruit : "banana"
▸ company : Object
▸ tags : Array (3)
```

Author Schema:

```
_id: 102
name : "Harper Lee"
birth_year : 1926
```

Book Schema:

```
_id: 1
title : "The Great Gatsby"
author_id : 100
genre : "Classic"
```

- Dhruv Rajput
- 2K21/EP/34

Q1. How many users are active?

```
1 [
2   {
3     $match: {
4       isActive: true
5     }
6   },
7   {
8     $count: 'activeUsers'
9   }
10 ]
```

PIPELINE OUTPUT

Sample of 1 document

activeUsers : 516

Q2. What is the average age of all users?

```
1 [
2   {
3     $group: {
4       _id: null, // null combines all data in 1 group
5       averageAge: {
6         $avg: "$age"
7       }
8     }
9   }
10 ]
```

PIPELINE OUTPUT

Sample of 1 document

averageAge : 29.835
_id: null

Q3. Find average age of Male and Female.

```
1 [
2   {
3     $group: {
4       _id: "$gender", // null combines all data in 1
5       averageAge: {
6         $avg: "$age"
7       }
8     }
9   }
10 ]
```

PIPELINE OUTPUT

Sample of 2 documents

_id: "female"
averageAge : 29.81854043392505

_id: "male"
averageAge : 29.851926977687626

Q4. List the top 2 most common fruits among the users.

```
1 [
2   {
3     $group: {
4       _id: "$favoriteFruit",
5       count: {
6         $sum: 1
7       }
8     },
9   },
10  {
11    $sort: {
12      count: -1 // -1 for descending
13    },
14  },
15  {
16    $limit: 2
17  }
18 ]
```

PIPELINE OUTPUT
Sample of 2 documents

OUTPUT OPTIONS ▾

```
_id: "banana"
count: 339
```

```
_id: "apple"
count: 338
```

Q5. Which country has the highest number of registered users?

```
1 [
2   {
3     $group: {
4       _id: "$company.location.country",
5       userCount: {
6         $sum: 1
7       }
8     },
9   },
10  {
11    $sort: {
12      userCount: -1
13    },
14  },
15  {
16    $limit: 1
17  }
18 ]
```

PIPELINE OUTPUT
Sample of 1 document

OUTPUT OPTIONS ▾

```
userCount : 261
_id: "Germany"
```

Q6. What is the average number of tags per user?

```
2 1
3 $unwind: "$tags", // create new doc for each tag
4 },
5 {
6   $group: {
7     _id: "$_id",
8     countOfTags: {
9       $sum: 1
10    }
11  },
12 },
13 {
14   $group: {
15     _id: null,
16     averageTags: {
17       $avg: "$countOfTags"
18     }
19  }
```

PIPELINE OUTPUT
Sample of 1 document

OUTPUT OPTIONS ▾

```
_id: null
averageTags : 3.556
```

OR

```
1 [
2   {
3     $addFields: {
4       numberOfTags: {
5         $size: {
6           $ifNull: ["$tags", []]
7         }
8       }
9     },
10  },
11  {
12    $group: {
13      _id: null,
14      averageNumberOfTags: {
15        $avg: "$numberOfTags"
16      }
17    }
18  }
19 ]
```

PIPELINE OUTPUT
Sample of 1 document

OUTPUT OPTIONS ▾

```
{
  "_id": null,
  "averageNumberOfTags": 3.556
}
```

Q7. How many users have 'enim' as one of their tags?

```
1 [
2   {
3     $match: {
4       tags: "enim"
5     }
6   },
7   {
8     $count: "usersWithEnimTag"
9   }
10 ]
```

PIPELINE OUTPUT
Sample of 1 document

OUTPUT OPTIONS ▾

```
{
  "usersWithEnimTag": 62
}
```

Q8. What are the names and age of the users who are inactive and have 'velit' as a tag?

```
1 [
2   {
3     $match: {
4       tags: "velit", isActive: false
5     }
6   },
7   {
8     $project: {
9       name: 1,
10      age: 1
11    }
12  }
13 ]
```

PIPELINE OUTPUT
Sample of 10 documents

OUTPUT OPTIONS ▾

```
{
  "name": "Aurelia Gonzales",
  "age": 20,
  "_id": ObjectId('66d42b02251a7de92d86f562')
}
```

```
{
  "age": 21,
  "_id": ObjectId('66d42b02251a7de92d86f580'),
  "name": "Hahn Pope"
}
```

```
{
  "age": 33,
  "_id": ObjectId('66d42b02251a7de92d86f58f')
}
```

Q9. How many users have a phone number starting with '+1 (940)'?

The image shows the MongoDB Compass interface. On the left, the aggregation pipeline is defined in a dark-themed editor. The pipeline consists of two stages: a \$match stage and a \$count stage. The \$match stage filters documents where the 'company.phone' field matches the regular expression '^\\+1 \\(940\\)'. The \$count stage counts the number of documents that match this criteria and labels the result as 'users with special number'. On the right, the 'PIPELINE OUTPUT' section shows a sample of 1 document, which is a single object containing the key 'users with special number' and the value 5. There is an 'OUTPUT OPTIONS' dropdown menu in the top right corner of the output section.

```
1 [
2   {
3     $match: {
4       "company.phone": /^\\+1 \\(940\\)/
5     }
6   },
7   {
8     $count: 'users with special number'
9   }
10 ]
```

PIPELINE OUTPUT
Sample of 1 document

users with special number : 5

Q10. Find the name, registration date and favorite fruit of 5 most recently registered users.

The image shows the MongoDB Compass interface. On the left, the aggregation pipeline is defined. It starts with a \$sort stage sorting documents by the 'registered' field in descending order (-1). This is followed by a \$limit stage limiting the results to 5 documents. Finally, a \$project stage is used to select only the 'name', 'registered', and 'favoriteFruit' fields from each document. On the right, the 'PIPELINE OUTPUT' section shows a sample of 5 documents. The first document shown is for 'Stephenson Griffith', registered on 2018-04-14, with 'apple' as a favorite fruit. The second document shown is for 'Sonja Galloway', registered on 2018-04-11, with 'strawberry' as a favorite fruit. There is an 'OUTPUT OPTIONS' dropdown menu in the top right corner of the output section.

```
1 [
2   {
3     $sort: {
4       registered: -1
5     }
6   },
7   {
8     $limit: 5
9   },
10  {
11    $project: {
12      name: 1,
13      registered: 1,
14      favoriteFruit: 1
15    }
16  }
17 ]
```

PIPELINE OUTPUT
Sample of 5 documents

_id: ObjectId('66d42b02251a7de92d86f824')
name : "Stephenson Griffith"
registered : 2018-04-14T03:16:20.000+00:00
favoriteFruit : "apple"

registered : 2018-04-11T12:52:12.000+00:00
favoriteFruit : "strawberry"
_id: ObjectId('66d42b02251a7de92d86f715')
name : "Sonja Galloway"

Q11. Categorize users by their favorite fruit.

The image shows the MongoDB Compass interface. On the left, the aggregation pipeline is defined. It uses a \$group stage to group documents by their 'favoriteFruit' field. For each group, it creates an array of user names using the \$push operator. The result is structured with '_id' as the group key and 'users' as an array of names. On the right, the 'PIPELINE OUTPUT' section shows a sample of 3 documents. The first document is for 'apple' with 338 users. The second is for 'banana' with 339 users. The third is for 'strawberry' with 323 users, and it shows the first two names in the array: 'Hays Wise' and 'Karyn Rhodes'. There is an 'OUTPUT OPTIONS' dropdown menu in the top right corner of the output section.

```
1 [
2   {
3     $group: {
4       _id: "$favoriteFruit",
5       users: {
6         $push: "$name"
7       }
8     }
9   }
10 ]
```

PIPELINE OUTPUT
Sample of 3 documents

users : Array (338)
_id: "apple"

_id: "banana"
users : Array (339)

_id: "strawberry"
users : Array (323)
0: "Hays Wise"
1: "Karyn Rhodes"

Q12. How many users have 'ad' as their second tag in their list of tags?

```
1 [
2 {
3   $match: {
4     "tags.1": "ad"
5   }
6 },
7 {
8   $count: "secondTagAd"
9 }
10 ]
```

PIPELINE OUTPUT

Sample of 1 document

secondTagAd : 12

Q13. Find users who have both 'enim' and 'id' in their tags.

```
1 [
2 {
3   $match: {
4     tags: {
5       $all: ['enim', 'id']
6     }
7 }
8 ]
9 ]
```

PIPELINE OUTPUT

Sample of 5 documents

```
registered : 2015-02-11T04:22:39.000+00:00
gender : "female"
favoriteFruit : "banana"
tags : Array (5)
eyeColor : "green"
company : Object
  _id: ObjectId('66d42b02251a7de92d86f562')
  index : 0
  name : "Aurelia Gonzales"
  isActive : false
  age : 20
```

Q14. List all companies located in USA with their corresponding user count.

```
1 [
2 {
3   $match: {
4     "company.location.country": "USA"
5   }
6 },
7 {
8   $group: {
9     _id: "$company.title",
10    userCount: {
11      $sum: 1
12    }
13 }
14 ]
15 ]
```

PIPELINE OUTPUT

Sample of 10 documents

```
_id: "SILODYNE"
userCount : 1
```

```
_id: "TOURMANIA"
userCount : 1
```

```
_id: "KIDSTOCK"
userCount : 1
```

Q15. Show details of author in books using concept of foreign key.

```
2 {
3   $lookup: {
4     from: "authors",
5     localField: "author_id",
6     foreignField: "_id",
7     as: "author_details_in_array"
8   },
9 },
10 { // now we want to take that object out from the
11   // array.
12   // basically taking the first element of the
13   // array
14 },
15 $addFields: {
16   author_details: {
17     $first: "$author_details_in_array"
18   }
19 }
20 }
```

PIPELINE OUTPUT

Sample of 3 documents

title: "The Great Gatsby"

author_id: 100

genre: "Classic"

author_details_in_array: Array (1)

author_details: Object

_id: 1

_id: 2

title: "Nineteen Eighty-Four"

author_id: 101

genre: "Dystopian"

author_details_in_array: Array (1)

OR

```
3 $lookup: {
4   from: "authors",
5   localField: "author_id",
6   foreignField: "_id",
7   as: "author_details_in_array"
8 },
9 },
10 { // now we want to take that object out from the
11   // array.
12   // basically taking the first element of the
13   // array
14 },
15 $addFields: {
16   author_details: {
17     $arrayElemAt: ["$author_details_in_array", 0]
18   }
19 }
20 }
21 }
```

PIPELINE OUTPUT

Sample of 3 documents

author_details: Object

_id: 1

title: "The Great Gatsby"

author_id: 100

genre: "Classic"

author_details_in_array: Array (1)

author_details_in_array: Array (1)

author_details: Object

_id: 2

title: "Nineteen Eighty-Four"

author_id: 101