

Documentation :

<https://www.mongodb.com/docs/manual/core/aggregation-pipeline/>

Random Data :

<https://gist.github.com/hiteshchoudhary>

\$match :

Return data with field that mention in match segment.

For example :

Aggregate_Demo.users

1k 1
DOCUMENTS INDEXES

Documents **Aggregations** Schema Indexes Validation

Pipeline **\$match** \$count Explain Export Run More Options ▾

Untitled - modified SAVE + CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT ✱ ⚙

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

PIPELINE OUTPUT
Sample of 1 document

ActiveUsers: 516

OUTPUT OPTIONS ▾

How many users are active :

In above it first return documents that have isActive : true and then it goes to input as in next stage and count returns number of documents.

Question 2 :

What is the average age of all users :

\$group : group all documents based on field name.

If you want to group all documents together then use "null".

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

PIPELINE OUTPUT
Sample of 1 document

_id: null
averageAge: 29.835

OUTPUT OPTIONS ▾

Group them by field gender.

The image shows the MongoDB Compass interface. On the left, a MongoDB aggregation pipeline is defined in a code editor. The pipeline consists of a single stage: `$group`, which groups documents by the `gender` field and calculates the average age for each group. The pipeline is as follows:

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

On the right, the 'PIPELINE OUTPUT' section shows a sample of 2 documents resulting from the aggregation:

```
{ "_id": "female", "averageAge": 29.81854043392505 }
{ "_id": "male", "averageAge": 29.851926977687626 }
```

`$group` : groups value based on unique values of all documents.

Question 3 :

List the top 5 most common fruits among all users :

`$sort` : sort the document based on count if you give 1 :lowest value at top ,
-1 : highest value at top.

`$limit` : returns number of documents from the result that mentioned.

The image shows the MongoDB Compass interface. On the left, a MongoDB aggregation pipeline is defined in a code editor. The pipeline consists of two stages: `$group` and `$sort`. The `$group` stage groups documents by the `favoriteFruit` field and calculates the count for each group. The `$sort` stage sorts the results in descending order of count. The pipeline is as follows:

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

On the right, the 'PIPELINE OUTPUT' section shows a sample of 3 documents resulting from the aggregation:

```
{ "_id": "banana", "count": 339 }
{ "_id": "apple", "count": 338 }
{ "_id": "strawberry", "count": 323 }
```

Accumulator : uses previous value and counts it.

Question 4 :

Find the total number of males and females :

```

1  [
2  {
3    $group: {
4      _id: "$gender",
5      genderCount: {
6        $sum: 1
7      }
8    }
9  }
10 ]

```

PIPELINE OUTPUT

OUTPUT OPTIONS

Sample of 2 documents

_id: "male"

genderCount: 493

_id: "female"

genderCount: 507

Counting is done through operator sum. We have to sum them because firstly we grouped them together.

Question 5 :

Which country has the highest number of registered users :

```

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

```

PIPELINE OUTPUT

OUTPUT OPTIONS

Sample of 1 document

_id: "Germany"

countUsers: 261

Question 6 :

List all unique eye colours present in the collection :

```

1  [
2  {
3    $group: {
4      _id: "$eyeColor"
5    }
6  }
7  ]

```

PIPELINE OUTPUT

OUTPUT OPTIONS

Sample of 3 documents

_id: "green"

_id: "blue"

_id: "brown"

Question 7 :

What is the average number of tags per users :

```

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

```

PIPELINE OUTPUT

Sample of 1 document

```

_id: null
averageNumberOfTags: 3.556

```

```

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

```

PIPELINE OUTPUT

Sample of 1 document

```

_id: null
averageNumberOfTags: 3.556

```

\$unwind : Deconstructs an array field from the input documents to output a document for *each* element.

\$addFields : Adds new fields to documents. `$addFields` outputs documents that contain all existing fields from the input documents and newly added fields.

Question 8 :

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

```

userswithenimtag: 62

```

Match : filters out the document.

Question 9 :

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

\$project : passes along the documents with the requested fields to the next stage in the pipeline.

The image shows the MongoDB Compass interface. On the left, a JSON document is displayed with line numbers 1 to 13. The document is:

```
{
  $match: {
    isActive: false, tags: 'velit'
  },
  $project: {
    name: 1,
    age: 1
  }
}
```

 On the right, the 'PIPELINE OUTPUT' section shows 'Sample of 29 documents'. It displays two document snippets:

```
{
  _id: ObjectId('65a79a78a85d139e67eb9e95'),
  name: 'Aurelia Gonzales',
  age: 20
}
```

 and

```
{
  _id: ObjectId('65a79a78a85d139e67eb9eb3'),
  name: 'Hahn Pope',
  age: 21
}
```

Question 10 :

How many users have a phone number starting with +1(940) :

The image shows the MongoDB Compass interface. On the left, a JSON document is displayed with line numbers 1 to 10. The document is:

```
{
  $match: {
    "company.phone" : /^+1 \ (940\)/
  },
  $count : "users with phone"
}
```

 On the right, the 'PIPELINE OUTPUT' section shows 'Sample of 1 document'. It displays a single document:

```
{
  users with phone: 5
}
```

Question 11 :

Who has registered most recently :

The image shows the MongoDB Compass interface. On the left, a JSON document is displayed with line numbers 2 to 17. The document is:

```
{
  $sort: {
    registered: -1
  },
  $limit: 4
},
$project: {
  name: 1,
  registered: 1,
  favoriteFruit: 1
}
```

 On the right, the 'PIPELINE OUTPUT' section shows 'Sample of 4 documents'. It displays two document snippets:

```
{
  _id: ObjectId('65a79a78a85d139e67eba157'),
  name: 'Stephenson Griffith',
  registered: 2018-04-14T03:16:20.000+00:00,
  favoriteFruit: 'apple'
}
```

 and

```
{
  _id: ObjectId('65a79a78a85d139e67eba048'),
  name: 'Sonja Galloway',
  registered: 2018-04-11T12:52:12.000+00:00,
  favoriteFruit: 'strawberry'
}
```

Question 12 :

Categorise users by their favourite fruit :

\$push : return an array by adding existing values to an array.

```

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

```

PIPELINE OUTPUT

Sample of 3 documents

_id: "banana"

users: Array (339)

_id: "apple"

users: Array (338)

_id: "strawberry"

users: Array (323)

Question 13 :

How many users have 'ad' as the second tag in their list of tags :

```

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

```

PIPELINE OUTPUT

Sample of 1 document

userswithadtag: 12

Question 14 :

Find users who have both 'enim' and 'id' as tag :

\$all : selects the document where the value of an array field contains specific elements.

```

1  [
2  {
3    $match: {
4      tags : {
5        $all : ['enim','id']
6      }
7    },
8  },
9  {
10   $project: {
11     name : 1
12   }
13 }
14 ]

```

PIPELINE OUTPUT

Sample of 5 documents

_id: ObjectId('65a79a78a85d139e67eb9e95')

name: "Aurelia Gonzales"

_id: ObjectId('65a79a78a85d139e67eb9ff1')

name: "Mcgowan Rosales"

_id: ObjectId('65a79a78a85d139e67eba0ab')

name: "Fisher Mercer"

Question 15 :

List all the companies located in the USA with their corresponding users count.

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

PIPELINE OUTPUT

Sample of 1 document

_id: "USA"
userCount: 255

OUTPUT OPTIONS ▾

Question 16 :