

1) Match

2) Count

Ques ➡ How many users are active ?

[

{

\$match: {

isActive: true,

},

},

{

// Number of users which are active ?

\$count: "NoofActiveUser",

},

]

Ques ➔ group the data based on gender

3) Group

```
[
  {
    //
    $group: {
      _id: "$gender"
    }
  }
]
```

4) Avg

Ques ➔ // group the data based on gender and count avg age

```
[
  {
    $group: {
      _id: "$gender",
      avgerageAge: {
        $avg: "$age"
      }
    }
  }
]
```

Ques → group the all data into one document and find the average age

```
[
  {
    $group: {
      _id: null,
      averageAge: {
        $avg: "$age"
      }
    }
  }
]
```

5) Match and Group

Ques → group the data based on age and find the age of 37

```
[
  {
    $group: {
      _id: "$age"
    },
  },
  {
    $match: {
      _id: 37
    }
  }
]
```

6) Sum

Ques → count Number of user for each age group

```
[
  {
    $group: {
      _id: "$age",
      count: { $sum: 1 }
    }
  }
]
```

Ques → count avgerage age of whole document

```
[
{
  $group: {
    // convert many doc to 1 single doc in order to count avg
    _id: null,
    AvgerageAge:{
      $avg:"$age"
    }
  }
}
]
```

7) Sort

Ques → group it by the fruits and sort it

```
[
  // stage 1
  {
    $group: {
      _id: "$favoriteFruit",
      count: {
        $sum: 1
      }
    }
  },

  // stage 2
  {
    $sort: {

      // count field is not available in original doc but
      // as we are in arrg pipeline hence the stage 1 output will be the new
      // input doc for stage 2
      count: -1

      // -1 : higher to lower
      // 1 : lower to higher
    }
  }
]
```

Ques → List the top 2 favorite fruit among all

```
[
  // stage 1
  {
    $group: {
      _id: "$favoriteFruit",
      count: {
        $sum: 1
      }
    }
  },

  // stage 2
  {
    $sort: {

      // count field is not available in original doc but
      // as we are in arrg pipeline hence the stage 1 output will be the new
      // input doc for stage 2
      count: -1
    }
  },
  {
    // give me top 2
    $limit: 2
  }
]
```

Ques ➔ Find the total number of males and females from high to low

```
[
  {
    $group: {
      _id: "$gender",
      count: {
        $sum: 1
      }
    }
  },
  {
    $sort: {
      count: -1 // desending order
    }
  }
]
```

Ques ➔ Which country has the highest number of registered user

```
[
  {
    $group: {
      _id: "$company.location.country",
      NoOfRegistereduser: {
        $sum: 1
      }
    },

  },
  {
    $sort: {
      NoOfRegistereduser: -1
    }
  },
  {
    $limit: 1
  }
]
```

Ques → what is avg number of tags per user

```
[
  {
    $unwind: {
      path: "$tags"
    }
  },
  {
    $group: {
      _id: "$_id",
      NumberOfTags: {
        $sum: 1
      }
    }
  },
  {
    $group: {
      _id: null,
      averageNumberOfTage: {
        $avg: "$NumberOfTags"
      }
    }
  }
]
```

UNWIND

Ques ➔ how many users does have 'enim' as one of their tags

```
[
  {
    $unwind: {
      path: "$tags",
    }
  },
  {
    $match: {
      tags: "enim"
    }
  },
  {
    $count: "NoOfenimUsers"
  }
]
```

Or

```
[
  {
    $match: {
      tags: "enim"
    }
  },
  {
    $count: "NoOfenimUser"
  }
]
```

PROJECT

Ques ➔ What are the name and age of users who are inactive and have 'velit' as a tag

```
[
  {
    $match: {
      tags: "velit",
      isActive : false
    }
  },
  {
    $project: {
      name:1,
      age:1
    }
  }
]
```

Ques ➔ how many users has phone number starting with +1 (940)

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

Ques → who has registered recently

```
[
  {
    $sort: {
      registered:-1
    }
  },
  {
    $limit:4
  },
  {
    $project: {
      name:1,
      age:1,
      registered:1
    }
  }
]
```

Ques ➔ Categorized users based on favorite fruit

PUSH

```
[
  {
    $group: {
      _id: "$favoriteFruit",
      users: {
        $push: "$name"
      }
    }
  }
]
```

Ques ➔ Categorized users based on favorite fruit and have name and age

```
[
  {
    $group: {
      _id: "$favoriteFruit",
      users: {
        $push: {
          name: "$name",
          age: "$age"
        }
      }
    }
  }
]
```

Ques ➔ how many users does have 'ad' as second position in tags

```
[
  {
    $match:{
      "tags.1" : "ad"
    }
  },
  {
    $count : "NoOfUsers"
  }
]
```

ALL

Ques ➔ Find users who have both 'enim' and 'id' as their tags

```
[
  {
    $match:{
      "tags":{
        $all:['enim' , 'id']
      }
    }
  },
]
```


Ques → list all companies which is located in USA with their corresponding user count

// aa total count che

```
[
  {
    $match: {
      "company.location.country" : "USA"
    }
  },
  {
    $project: {
      "company.location.country" : 1
    }
  },
  {
    $count: "NoOfUsers"
  }
]
```

```
// with their corresponding user count
[
  {
    $match: {
      "company.location.country" : "USA"
    }
  },
  {
    $group: {
      _id: "$company.title",
      count : {
        $sum: 1
      }
    }
  }
]
```

Ques → JOINS in mongodb is done through “**LOOKUP**”

```
[
  {
    $lookup: {
      from: "authors",
      localField: "author_id",
      foreignField: "_id",
      as: "author_details"
    }
  },
  {
    $addFields: {
      authorDetails: {
        $arrayElemAt: ["$author_details",0]
      }
    }
  }
]
```

INNER JOIN

```
[
  {
    $lookup: {
      from: "authors",
      localField: "author_id",
      foreignField: "_id",
      as: "author_details"
    }
  },
  {
    $addFields: {
      authorDetails: {
        $arrayElemAt: ["$author_details",0]
      }
    }
  },
  {
    $match: {
      "authorDetails.birth_year" : 1896
    }
  }
]
```

OR

```
[
  {
    $lookup: {
      from: "authors",
      localField: "author_id",
      foreignField: "_id",
      as: "author_details"
    }
  },
  {
    $unwind: "$author_details"
  },
  {
    $match: {
      "author_details.birth_year": 1896
    }
  }
]
```

Logical Operators in MongoDB in Hindi (\$not, \$and, \$or & \$nor)

```
college> db.students.find({ $or:[ {age:{$lte:10}} , {age:{$lte:12}} ]})
```

```
[
  {
    _id: ObjectId("6396a083f3b30b3e904d794f"),
    name: 'Sita',
    age: 5
  }
]
```

Logical Operators in MongoDB in Hindi (\$not, \$and, \$or & \$nor)

```
college> db.students.find({ $nor:[ {age:{$lte:10}} , {age:{$lte:12}} ]})
```

```
[
  {

```

Logical Operators in MongoDB in Hindi (\$not, \$and, \$or & \$nor)

```
college> db.students.find({ $and:[ { age:{$lt:11}} , { Hobbies:'Walk' } ] })
```

```
[
  {
    _id: ObjectId("6396a083f3b30b3e904d794f"),
    name: 'Sita',
    age: 5,
    hobbies: [ 'Walk', 'Cricket' ],
    identity: { hasPanCard: false, hasAdhaarCard: true }
  }
]
```

Logical Operators in MongoDB in Hindi (\$not, \$and, \$or & \$nor)

```
college> db.students.find({ $and:[ {age:{$lt:11}}, { age:{$gt:5}} ] })
```

```
[
  {

```

Logical Operators in MongoDB in Hindi (\$not, \$and, \$or & \$nor)

```
college> db.students.find({ $and:[ {age:{$lt:11}}, { age: { $not: { $gt:5 } } } ] })
```

\$eq \$ne

\$lt \$gt

\$lte \$gte

\$in \$nin

\$and

\$or

\$nor

\$not