School mongodb

1 . insert data to collection

```
db.school.insertMany( [
{
    "name": "mokbul hossain",
    "age" : 29,
    "class" : 9,
    "status" : "pass",
    "position" : 10,
    "bangla": 80,
    "math": 55,
    "marks": 135,
    "gpa": 4
},
{
    "name": "imran ahmed",
    "age" : 32,
    "class" : 9,
    "status" : "pass",
    "position" : 12,
    "bangla": 40,
    "math": 30,
    "marks": 70,
    "gpa": 3.38
}
]
)
```

Or

```
db.school.insertMany( [
{
    "name": "mokbul hossain",
    "age" : 30,
    "class" : 10,
    "status" : "pass",
    "position" : 2,
    "subject": {
      "bangla":100,
      "math":100
    },
    "marks": 200,
    "gpa": 5
```

```
},
{
    "name": "imran ahmed",
    "age" : 34,
    "class" : 10,
    "status" : "pass",
    "position" : 6,
    "subject": {
      "bangla":20,
      "math":20
    },
    "marks": 200,
    "gpa": 3.45
}
]
)
```

2 .

```
db.getCollection('school').updateMany(
{},
{ $set: { date: new Date() } }

)
```

3 . and operation

```
db.getCollection('school').find(
{ gpa: {$gt:4 } , marks: { $lt: 100 } },
{ name: 1 , class: 1 , marks:1 }

)
```

4 . or operation

```
db.getCollection('school').find(
{ $or: [ { gpa: {$gt:4 } } , { marks: { $lt: 100 } } ] },
{ name: 1 , class: 1 , marks:1 }

)
```

5 . find a text in a collection

```
db.getCollection('school').find(
{ name:/rah/ }
```

```
)

db.getCollection('school').find(
{ name:  { $regex : /rah/ } }
)
```

6 . sort

```
db.school.find( {} )
.sort( { marks : -1 } )

db.school.find( { name: /ra/ } )
.sort( { marks : -1 } )
```

7 . count documents

```
db.school.find( {} )
.count()

db.school.find( { name: /ra/ } )
.sort( { marks : -1 } )
.count()
```

8 . update

```
db.school.updateMany(
 { age:34.0 },
 { $set: { position: 8.0 } }
  )
```

9 . find documents

```
db.school.find(
 {},
 { "subject.math" : 1 }
  )
```

Or for all subject

```
db.school.find(
 {},
 { subject : 1 }

  )
```

10 . aggregate operations

For distinct name

```
db.school.aggregate( [
 {
  $group: { _id:"$name" }
 }

  ]
 )
```

```
db.school.aggregate( [
   {
      $group: {
            _id:null,
            count:{ $sum:1 }
           }
   }
] )
```

```
const data = await stuData.aggregate(
            [
                {
                    $group:
                    {
                          _id: "$age" ,
                          count:{ $sum:1 }
                    }
                }
            ]
        )
```

Name wise total marks

```
db.school.aggregate( [
   {
      $group: {
            _id:"$name",
            count:{ $sum: "$marks" }
           }
   }
```

```
] )
```

Only total marks

```
db.school.aggregate( [
   {
      $group: {
            _id:null,
            count:{ $sum: "$marks" }
           }
   }
] )


db.school.aggregate(
,
)
```

Sort by name and get the summation of marks :

```
db.school.aggregate(
[
   {
      $group: {
         _id: "$name",
         summation : { $sum: "$marks" }
      }
   },
   { $sort: { summation: -1 } }

]
)

db.school.aggregate( [
{ $match : {  marks: { $gt:100 } }  },
{
   $group:
   {
     _id:"$name",
     count : { $sum:1 }
   }
}
] )
```

Name which matches the regular expression rah

```
db.school.aggregate( [
{ $match : {  name: /rah/ }  },
{
   $group:
   {
     _id:"$name",
     count : { $sum:1 }
   }
}
] )
```

Aggregate the name and the summation of marks to total and then match the total if it is greater than 150 .

```
db.school.aggregate( [
   {
      $group:
      {
        _id:"$name",
        total : { $sum:"$marks" }
      }
   },
   { $match : { total : { $gt: 150 } }  }

] )
```

The unwind operations

```
db.school.insertOne(
{  sizes: [ "S", "M", "L"] })


db.school.aggregate(
[
{ $unwind: "$sizes" }
]
)
```

Sortbycount operations

```
db.school.insertOne(
{
  "tags" : [ "abstract", "painting" ]

    })
```

We can find specific tag total number

```
db.school.aggregate( [
{  $unwind: "$tags"    } ,
{ $sortByCount: "$tags" }
]
)
```

Another way we find
The group and the sort keyword in aggregation

```
db.school.aggregate( [
{  $unwind: "$tags"    } ,
{
   $group: {
       _id:"$tags",
       count: { $sum:1 }
   }
},
{ $sort : { count: -1 } }
]
)

db.school.insertOne(
{
 "quizzes": [ 4, 5, 5 ], "labs": [ 6, 5 ], "final": 78, "midterm": 70
})
```

Using the project procedures

```
db.school.aggregate([
  {
    $project: {
      quizTotal: { $sum: "$quizzes"},
      labTotal: { $sum: "$labs" },
      examTotal: { $sum: [ "$final", "$midterm" ] }
    }
  }
])

/* 18 */
{
    "_id" : ObjectId("6061b0f87e406343269d7fba"),
    "quizTotal" : 14.0,
    "labTotal" : 11.0,
    "examTotal" : 148.0
}
```

Using the group procedures

```
db.school.aggregate([
 {  $unwind: "$quizzes"} ,
 {
    $group: {
      _id:"$_id",
      sum: { $sum: "$quizzes" }

    }
  }
])

/* 1 */
{
    "_id" : ObjectId("6061b0e07e406343269d7fb8"),
    "sum" : 23.0
}

/* 2 */
```

```
{
    "_id" : ObjectId("6061b0ec7e406343269d7fb9"),
    "sum" : 19.0
}

/* 3 */
{
    "_id" : ObjectId("6061b0f87e406343269d7fba"),
    "sum" : 14.0
}
```

10 . sort by two field .

```
db.school.aggregate([

 { $sort: { marks: -1 , gpa: 1 } }

])
```

11 . summation and multiplication of bangla and math and then sum the multiplied values

```
db.school.aggregate(
  [

    {
      $group :
        {
          _id : "$math",
          totalSaleAmount: { $sum: { $multiply: [ "$bangla", "$math" ] } }
        }
     },
     { $sort: { totalSaleAmount: -1 } }

  ]
 )
```

12 . lookup operations between school and teacher collection .

```
db.school.aggregate(
[
{
  $lookup:
  {
    from: "teacher",
    localField: "class",
    foreignField: "cls",
    as : "haha"
  }
}
] )

/* 2 */
{
    "_id" : ObjectId("606181e37e406343269d7faa"),
    "name" : "aminul islam",
    "age" : 25.0,
    "class" : 9.0,
    "status" : "pass",
    "position" : 7.0,
    "bangla" : 70.0,
    "math" : 70.0,
    "marks" : 140.0,
    "gpa" : 4.1,
    "date" : ISODate("2021-03-29T07:33:19.382Z"),
    "haha" : [
        {
            "_id" : ObjectId("6061c0547e406343269d7fbb"),
            "name" : "monir",
            "cls" : 9.0
        }
    ]
}
```

13 . match the marks with or operations and group the documents then count the data.

```
db.school.aggregate(
[
    {
      $match: {
          $or:
          [
             { marks:  { $gt: 80 , $lt: 120 }  } ,
             { gpa : { $gt: 4 } }
          ]
        }
     },
    {
       $group: {
          _id:null,
         count: { $sum:1 }

       }

    }


] )
```